บทที่ 2

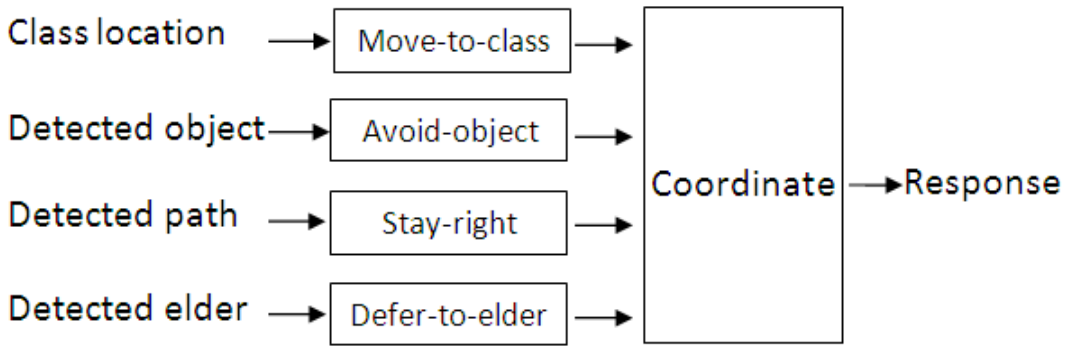
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

การวิจัยเรื่องการพัฒนาระบบการเคลื่อนที่ของหุ่นยนต์ด้วยฟัซซีลอจิก ผู้วิจัยได้ตรวจสอบเอกสารและผลงานวิจัยที่เกี่ยวข้องนำเสนอตามลำดับดังนี้คือพฤติกรรมของหุ่นยนต์, สถาปัตยกรรมแบบระดับชั้น (Sub gumption architecture), ไมโครคอนโทรลเลอร์ PIC 16F877 และการทำงาน, คำสั่งภาษาเบสิกของไมโครคอนโทรลเลอร์ PIC 16F877, ซอฟแวร์ที่เกี่ยวข้องกับการใช้งาน การพัฒนาและประยุกต์ใช้งานไมโครคอนโทรลเลอร์, ตรรกศาสตร์คลุมเครือและผลงานวิจัยที่เกี่ยวข้อง

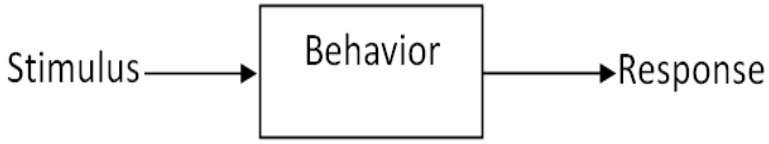
2.1 พฤติกรรมของหุ่นยนต์

ในการออกแบบระบบหุ่นยนต์สิ่งที่สำคัญคือการใส่พฤติกรรมให้กับหุ่นยนต์ เช่นการเคลื่อนที่ การหยุดการหลบหลีกสิ่งกีดขวาง ซึ่งพฤติกรรมพื้นฐานเหล่านี้ จะต้องมีสำหรับหุ่นยนต์ ดังนั้นเพื่อให้เข้าใจรูปแบบพฤติกรรมของหุ่นยนต์ เราสามารถเขียนรูปแบบแสดงพฤติกรรมได้ดังนี้

การแสดงพฤติกรรมของหุ่นยนต์ แผนภาพ SR (Stimulus-Response Diagram) แผนภาพแบบง่ายที่แสดงพฤติกรรม ดังภาพที่ 2-2 ซึ่งเป็นการแสดงพฤติกรรมผลตอบสนองต่อสิ่งกระตุ้นของพฤติกรรมหนึ่งๆ ภาพที่ 2-1 แสดงแผนภาพ SR สำหรับการนำทางเพื่อทำการตัดสินใจ เมื่อนำเอาต์พุตของแต่ละพฤติกรรมรวมกันจะได้ผลตอบสนองรวมสำหรับหุ่นยนต์ขณะช่วงเวลาและสภาพแวดล้อมหนึ่งๆ



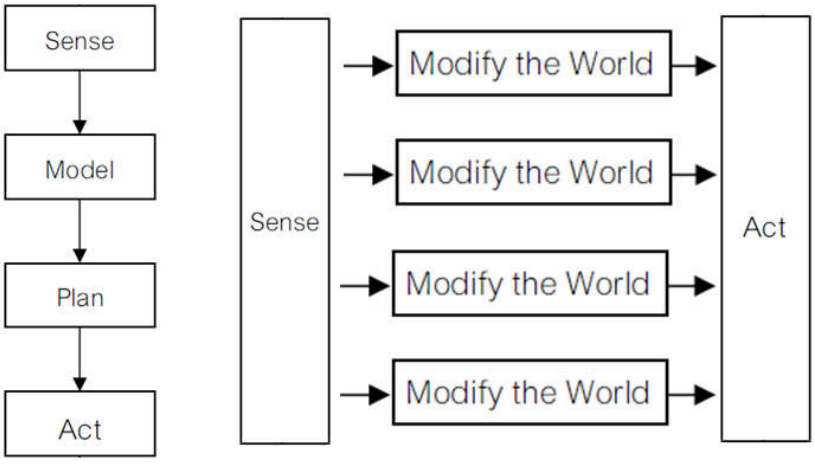
ภาพที่ 2-1 แผนภาพ SR สำหรับการนำทางเพื่อทำการตัดสินใจ



ภาพที่ 2-2 แผนภาพ SR ของพฤติกรรม

2.2 สถาปัตยกรรมแบบระดับชั้น (Sub sumption architecture)

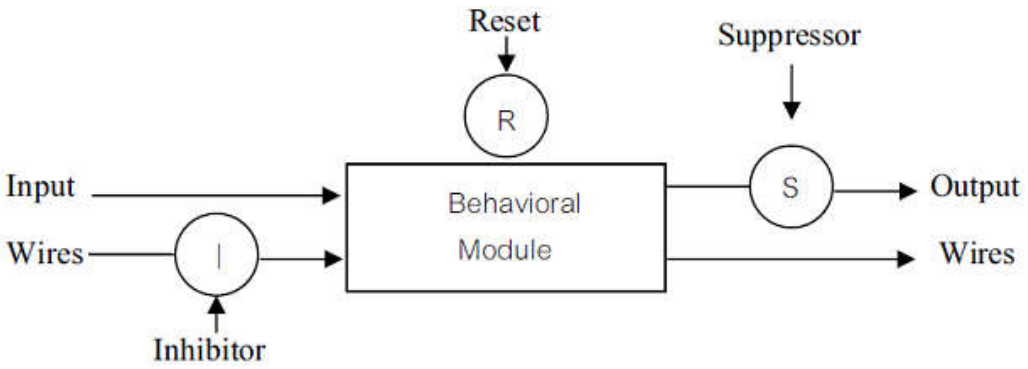
ศาสตราจารย์บรูคส์ *(Lester, Bruce P., 1980)* ได้พัฒนาสถาปัตยกรรมแบบระดับชั้น เมื่อกลางปี ค.ศ. 1980 ที่ Massachusetts Institute of Technology และมีการสัมมนาของศาสตราจารย์บรูคส์ ในปี ค.ศ. 1986 ได้กล่าว ถึงระบบควบคุมแบบระดับชั้นที่มีมิติต่างกันดังภาพที่ 2-3



(ก) (ข)

ภาพที่ 2-3 ระบบควบคุม (ก) รูปแบบปกติแนวตั้ง (ข) รูปแบบใหม่แนวนอน

พฤติกรรมในสถาปัตยกรรมแบบระดับชั้น จะแบ่งแยกลำดับชั้นแต่ละลำดับชั้นจะมีวัตถุประสงค์แยกจากกันแต่ละพฤติกรรมแสดงโดยใช้รูปแบบเอเอฟเอสเอ็ม (AFSM, Augmented Finite State Machine) ดังภาพที่ 2-4 AFSM ประกอบด้วยพฤติกรรม สัญญาณกระตุ้นและสัญญาณตอบสนอง ซึ่งสามารถระงับหรือยับยั้งโดยพฤติกรรมอื่น สัญญาณรีเซตใช้คืนพฤติกรรมสู่สภาวะเริ่มต้น



ภาพที่ 2-4 AFSM ในสถาปัตยกรรมแบบระดับชั้น

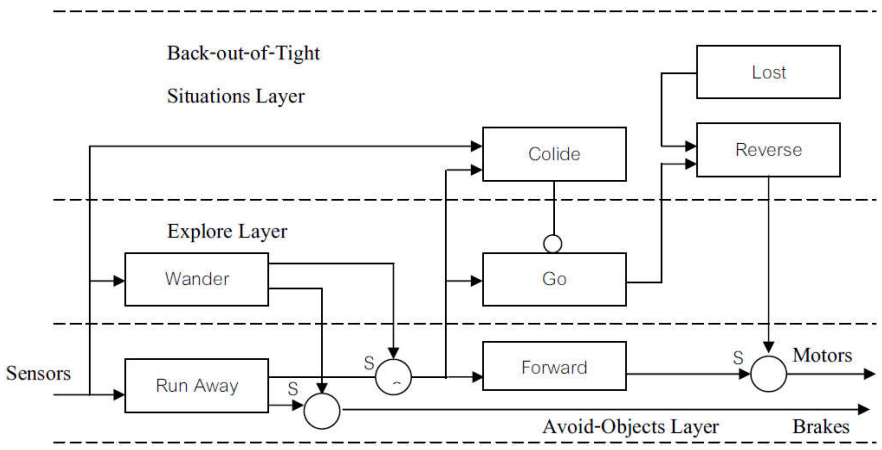
การออกแบบระบบโดยใช้สถาปัตยกรรมแบบระดับชั้นมีการใช้งานกว้างขวาง แต่ในปี ค.ศ. 1998 ฟอกซ์, เฮกมัวและมาตาริค *(Fox, D., Hexmoor, H. and Mataric, M., 1998)* ได้แสดงการออกแบบและพัฒนาในการกำหนดงานของหุ่นยนต์ ซึ่งมีขั้นตอนดังนี้

กำหนดคุณสมบัติของพฤติกรรมที่ต้องการสำหรับงาน

1. แบ่งและกำหนดพฤติกรรมอิสระของแต่ละคุณสมบัติของพฤติกรรม

2. กำหนดพฤติกรรมย่อยๆ และจัดวางแบบแยกชั้น โดยการตรวจจับสัญญาณและการเคลื่อนที่จะอยู่ที่พฤติกรรมระดับล่าง

การสร้างหุ่นยนต์ด้วยสถาปัตยกรรมแบบระดับชั้นอย่างกว้างขวางตัวอย่างเช่น หุ่นยนต์ Allen โดยบรูคส์ในปี ค.ค. 1990, หุ่นยนต์ Tom และ Jerry โดยโคนแนล (Connel) ในปี ค.ศ. 1987, หุ่นยนต์ Herbert โดยบรูคส์ในปี ค.ศ. 1988 และหุ่นยนต์ Genghis โดยบรูคส์ในปี ค.ศ. 1989 เป็นต้น

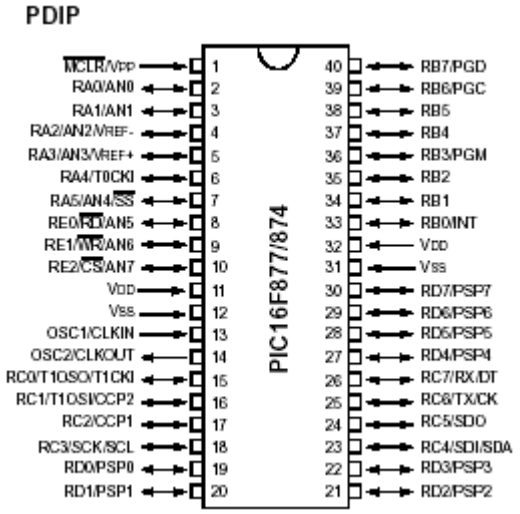


ภาพที่ 2-5 แสดงการประยุกต์ใช้ระบบระดับชั้นสำหรับหุ่นยนต์เคลื่อนที่แบบ 3 ลำดับ

ในภาพที่ 2-5 แสดงการประยุกต์ใช้ระบบระดับชั้นสำหรับหุ่นยนต์เคลื่อนที่แบบ 3 ลำดับชั้นตามพฤติกรรมระดับต่ำที่สุดเป็นพฤติกรรมการหลบสิ่งกีดขวางมีหน้าที่เลี้ยวหรือหยุด เพื่อหลบสิ่งกีดขวาง ขึ้นอยู่กับสัญญาณอินพุตจากตัวตรวจวัดอินฟราเรด พฤติกรรมระดับถัดมาเป็นพฤติกรรมการเคลื่อนที่สำรวจมีหน้าที่สั่งให้เคลื่อนที่ไปในทิศทางที่ไม่มีสิ่งกีดขวางอยู่ พฤติกรรมระดับสูงสุดเป็นพฤติกรรมการถอนตัวจากที่คับขันมีหน้าที่สั่งให้เคลื่อนที่กลับทิศทางในกรณีที่การหลบหลีกสิ่งกีดขวางและการเคลื่อนที่สำรวจไม่สามารถทำงานได้

2.3 ไมโครคอนโทรลเลอร์ PIC 16F877 และการทำงาน

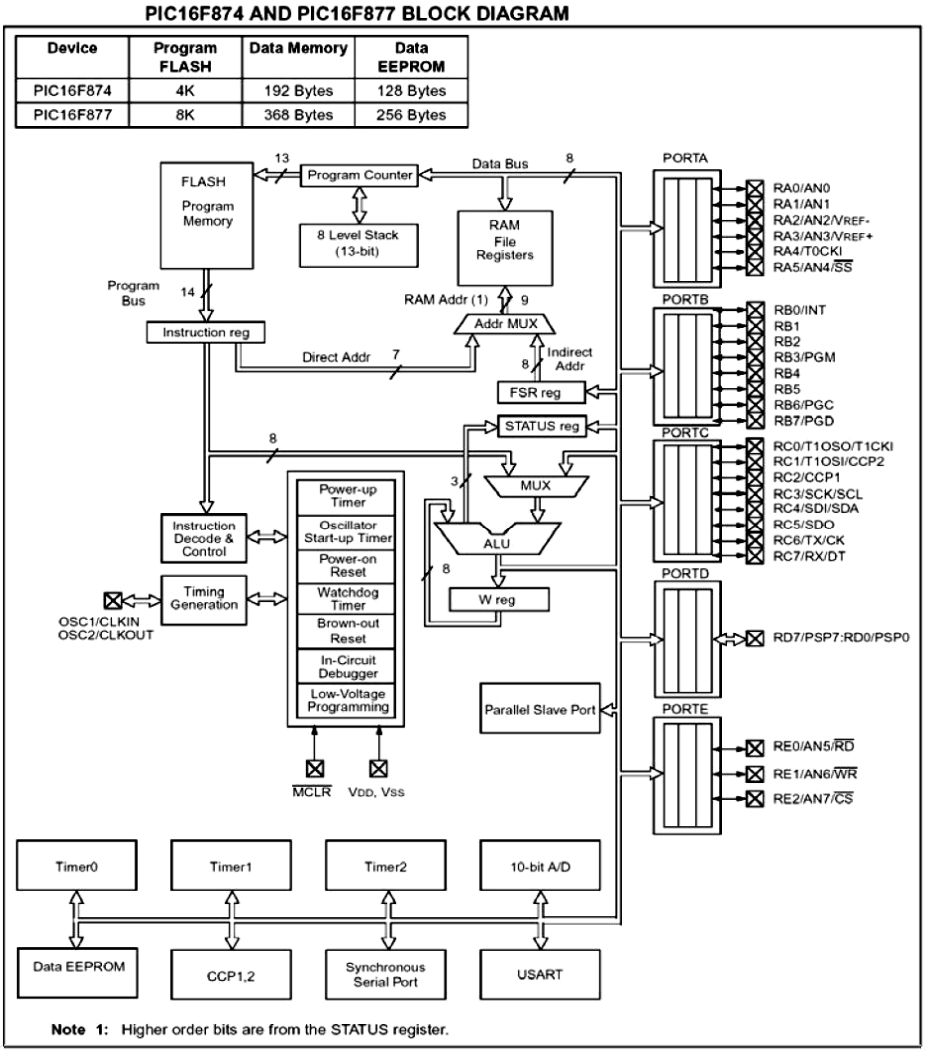
พื้นฐานการทำงานของไมโครคอนโทรลเลอร์ก็ คือ ระบบดิจิตอล โดยค่าเอาท์พุตที่ได้จากไมโครคอนโทรลเลอร์จะเป็น 0 กับ 1 แต่ก็สามารถนำมาประยุกต์เชื่อมต่อกับอุปกรณ์ภายนอกต่างๆ มากมาย



ภาพที่ 2-6 *ขาสัญญาณและหน้าที่การนำไปใช้งาน*

จากภาพที่ 2-6 *ขาของ* PIC*16f877 กับ 16*F*874 เหมือนกันแต่ความสามารถบางประการจะไม่เหมือนกัน สามารถดูได้จาก* data sheet (*ภาคผนวก ง) แต่ละขาจะมีหน้าที่แตกต่างกันไปซึ่งแยกออก เป็น* PORT A , PORT B , PORT C , PORT D *และ* PORT E *โดยพื้นฐานแล้วพอร์ตแต่ละพอร์ตสามารถทำงานเป็นอินพุตและเอาท์พุตเป็นดิจิตอล ยกเว้น* PORT A *และ* PORT E *ที่สามารถทำงานเป็นตัวรับสัญญาณอนาลอกแปลงเป็นค่าดิจิตอลเพื่อนำมาวัดปริมาณทางฟิสิกส์ต่างๆ ที่เห็นได้อย่างชัดเจน คือ นำมาวัดความต่างศักย์ (วัชรินทร์เการพ, 2545:1-8)*

ไมโครคอนโทรลเลอร์ตระกูล PIC *มีสถาปัตยกรรมแบบฮาร์วาร์ด (*Harvard architecture) *กล่าวคือมีการแยกหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลออกจากกันโดยมีบัสสำหรับติดต่อแยกกันด้วย ดังแสดงในภาพที่* 2-7 *จะเห็นว่า ซีพียูภายในไมโครคอนโทรลเลอร์ จะติดต่อกับหน่วยความจำโปรแกรมด้วยบัสแอดเดรส* 13 *บิต และบัสข้อมูลหน่วยความจำโปรแกรม* 14 *บิต ในขณะที่บัสสำหรับติดต่อหน่วยความจำข้อมูลและรีจิสเตอร์ ภายในเป็นแบบ* 8 *บิตทั้งบัสแอดเดรสและบัสข้อมูล*



ภาพที่ 2-7 *สถาปัตยกรรมภายในของ* PIC 16F874 *และ*16F877

นอกจากการจัดสถาปัตยกรรมแบบนี้แล้ว การกระทำคำสั่งของไมโครคอนโทรลเลอร์ PIC *ยังใช้กระบวนการที่เรียกว่า ไปป์ไลน์ (* Pipeline) *ทำให้สามารถเฟตช์คำสั่งถัดไป ในขณะที่กำลังทำ เอ็กซิคิวต์คำสั่งในปัจจุบัน ส่งผลให้ความเร็วในการทำงานของไมโครคอนโทรลเลอร์เพิ่มมาก*

ขึ้นนั่นจึงเป็นที่มาของความสามารถในการทำคำสั่ง 1 *คำสั่ง ภายในสัญญาณนาฬิกา* 1 *ลูก (เฟตช์ :* fetch *เป็นกระบวนการเรียกคำสั่งออกจากหน่วยความจำโปรแกรม แล้วแปลเป็นเลขฐานสิบหก เพื่อให้ ซีพียูเข้าใจ ส่วนกระบวนการเอ็กซิคิวต์* :Execute *เป็นการกระทำคำสั่งให้เกิดผลลัพธ์ ตามที่คำสั่งนั้นๆ กำหนด) คำสั่งต่างๆของไมโครคอนโทรลเลอร์ตามตารางที่* 2-1

ตารางที่ 2-1 *หน้าที่ขาสัญญาณของ* PIC16F877

ขาสัญญาณ	DIP PIN	Input/ Output Type	B u f f e r type	Description
OSC1/CLKIN	13	I	ST/CMOS	- ขาสัญญาณอินพุตสัญญาณนาฬิกา ของ CPU
OSC2/CLKOUT	14	O	-	- ขาเอาต์พุตของสัญญาณนาฬิกา ใช้ต่อร่วมกับขาสัญญาณ OSC1 เพื่อกำเนิดสัญญาณนาฬิกาในกรณีที่ใช้ คริสตอลแบบเรโซเนเตอร์หรือวงจร RC จากภายนอก
MCLR/VPP	1	I/P	ST	- ขาสัญญาณรีเซ็ทแอคทีฟ “0” ขารับ แรงดันสำหรับการโปรแกรม
RA0/AN0	2	I/O	TTL	- ขาสัญญาณอินพุต/เอาต์พุต PORT A - ขาอินพุตสัญญาณแอนะลอก AN0
RA0/AN1	3	I/O	TTL	- ขาสัญญาณอินพุต/เอาต์พุต PORT A - ขาอินพุตสัญญาณแอนะลอก AN1
RA2/AN2/VREF-	4	I/O	TTL	- ขาสัญญาณอินพุต/เอาต์พุต PORT A - ขาอินพุตสัญญาณแอนะลอก AN2 - ขาสัญญาณแรงดันอ้างอิงลบ(-)

ตารางที่ 2-1 (ต่อ)

RA2/AN3/VREF+	5	I/O	TTL	- ขาสัญญาณอินพุต/เอาต์พุต PORT A - ขาอินพุตสัญญาณแอนะลอก AN3 - ขาสัญญาณแรงดันอ้างอิงบวก(+)
RA4/TOCKI	6	I/O	TTL	- ขาสัญญาณอินพุต/เอาต์พุต PORT A - ขาอินพุตสัญญาณนาฬิกาของไทมเมอร์ 0
RA5/AN4/SS	7	I/O	TTL	- ขาสัญญาณอินพุต/เอาต์พุต PORT A - ขาอินพุตสัญญาณแอนะลอก AN4 - สัญญาณ Slave Select ในโหมดการสื่อสารแบบ Synchronous Serial Port
RB0/INT	33	I/O	TTL	- ขาสัญญาณอินพุต/เอาต์พุต PORT B - ขาอินพุตสัญญาณอินเตอร์รัพท์ภายนอก
RB1	34	I/O	TTL	- ขาสัญญาณอินพุต/เอาต์พุต PORT B
RB2	35	I/O	TTL	- ขาสัญญาณอินพุต/เอาต์พุต PORT B
RB3/PGM	36	I/O	TTL	- ขาสัญญาณอินพุต/เอาต์พุต PORT B - ขาสัญญาณการโปรแกรมแบบแรงดันต่ำ
RB4	37	I/O	TTL	- ขาสัญญาณอินพุต/เอาต์พุต PORT B - สัญญาณการอินเตอร์จากการเปลี่ยนแปลงสถานะของขาสัญญาณ
RB5	38	I/O	TTL	- ขาสัญญาณอินพุต/เอาต์พุต PORT B - สัญญาณการอินเตอร์จากการเปลี่ยนแปลงสถานะของขาสัญญาณ
RB6/PGC	39	I/O	TTL/ST	- ขาสัญญาณอินพุต/เอาต์พุต PORT B - สัญญาณการอินเตอร์จากการเปลี่ยนแปลงสถานะของขาสัญญาณ - ขาสัญญาณนาฬิกาในโหมดการเขียนโปรแกรม

ตารางที่ 2-1 (ต่อ)

RB7/PGD	40	I/O	TTL/ST	- ขาสัญญาณอินพุต/เอาต์พุต PORT B - สัญญาณการอินเตอร์จากการเปลี่ยนแปลงสถานะของขาสัญญาณ - ขาสัญญาณนาฬิกาในโหมดการเขียนโปรแกรม
RC0/T1OSO/TICKI	15	I/O	ST	- ขาสัญญาณอินพุต/เอาต์พุต PORT C - ขาสัญญาณออสซิลเลเตอรเอาต์พุตและ - ขาอินพุตสัญญาณนาฬิกาของไทม์เมอร์ 1
RC1/T1OSI/OCP2	16	I/O	ST	- ขาสัญญาณอินพุต/เอาต์พุต PORT C - ขาสัญญาณออสซิลเลเตอรเอาต์พุตและ - ขาอินพุตสัญญาณนาฬิกาของไทม์เมอร์ 1 - ขาสัญญาณ Capture2 input/Compare2 output/PWM2 output
RC2/CCP1	17	I/O	ST	- ขาสัญญาณอินพุต/เอาต์พุต PORT C - ขาสัญญาณออสซิลเลเตอรเอาต์พุตและ - ขาอินพุตสัญญาณนาฬิกาของไทม์เมอร์ 1 - ขาสัญญาณ Capture1 input/Compare2 output/PWM1 output
RC4/SDI/SDA	23	I/O	ST	- ขาสัญญาณอินพุต/เอาต์พุต PORT C - ขาอินพุตสัญญาณข้อมูลในโหมด SPI - ขาอินพุต/เอาต์พุตสัญญาณข้อมูลในโหมด I2C

ตารางที่ 2-1 (ต่อ)

RC6/TX/CK	25	I/O	ST	- *ขาสัญญาณอินพุต/เอาต์พุต* PORT C - *ขาเอาต์พุตสัญญาณด้านส่งของการสื่อสาร* USART - *ขาสัญญาณนาฬิกาในโหมดการสื่อสาร* Synchronous Description
RC7/TX/CK	26	I/O	ST	- *ขาสัญญาณอินพุต/เอาต์พุต* PORT C - *ขาเอาต์พุตสัญญาณด้านรับของการสื่อสาร* USART - *ขาสัญญาณนาฬิกาในโหมดการสื่อสาร* Synchronous Description
RD*0*/PSPO	19	I/O	ST/TTL	- *ขาสัญญาณอินพุต/เอาต์พุต* PORT D - ขา Slave Port *กรณีติดต่อกับระบบบัสของไมโครคอนโทรลเลอร์*
RD0/PSP1	20	I/O	ST/TTL	- *ขาสัญญาณอินพุต/เอาต์พุต* PORT D
RD0/PSP2	21	I/O	ST/TTL	- *ขาสัญญาณอินพุต/เอาต์พุต* PORT D
RD0/PSP3	22	I/O	ST/TTL	- *ขาสัญญาณอินพุต/เอาต์พุต* PORT D
RD0/PSP4	27	I/O	ST/TTL	- *ขาสัญญาณอินพุต/เอาต์พุต* PORT D
RD0/PSP5	28	I/O	ST/TTL	- *ขาสัญญาณอินพุต/เอาต์พุต* PORT D
RD0/PSP6	29	I/O	ST/TTL	- *ขาสัญญาณอินพุต/เอาต์พุต* PORT D
RD0/PSP7	30	I/O	ST/TTL	- *ขาสัญญาณอินพุต/เอาต์พุต* PORT D
RE0/RD/AN5	8	I/O	ST/TTL	- *ขาสัญญาณอินพุต/เอาต์พุต* PORT E - *ขาสัญญาณควบคุมการอ่านในโหมด* Parallel Slave Port - *ขาสัญญาณอินพุตสัญญาณแอนะลอก* AN5
RE1/WR/AN6	9	I/O	ST/TTL	- *ขาสัญญาณอินพุต/เอาต์พุต* PORT E - *ขาสัญญาณควบคุมการเขียนในโหมด* PSP

ตารางที่ 2-1 (ต่อ)

RE2/CS/AN7	10	I/O	ST/TTL	- *ขาสัญญาณอินพุต/เอาต์พุต* PORT E - *ขาสัญญาณควบคุมการติดต่อใน โหมด* Parallel Slave Port - *ขาสัญญาณอินพุตสัญญาณแอนะลอก* AN7
VSS	*12*, *31*	p	-	- *ขาสัญญาณกราวด์*
VDD	11, 32	p	-	-*ขาแรงดันไฟเลี้ยง*

หมายเหตุ : I คือ อินพุต

O คือ เอาต์พุต

I /O คือ อินพุต / เอาต์พุต

P คือ Power

- คือ ไม่ใช้

TTL คือ ระดับสัญญาณ TTL

ST คือ วงจรชมิตทริกเกอร์อินพุต

2.4 คำสั่งภาษาเบสิกของไมโครคอนโทรลเลอร์ PIC 16F877

2.4.*1* Comments *ใช้สำหรับอธิบายขั้นตอนการทำงานของโปรแกรมในแต่ละบรรทัดคำสั่ง ใช้เครื่องหมาย ฝนทอง (*Quotation mark) ‘ *หรือ* ; *เช่น* HIGH PORT B.*0* ‘Turn ON LED

2.4.*2* Line Labels *ใช้สำหรับกำหนดข้อความ (ไม่เกิน 8 ตัว) เพื่อใช้อ้างอิงตำแหน่งของคำสั่งที่จะย้อนกลับไปทำงานซ้ำ เนื่องจาก* PIC BASIC *เป็นรูปแบบที่ไม่มีบรรทัดคำสั่งที่เป็นตัวเลข ข้อความที่กำหนดเป็น* Label *ต้องมีเครื่องหมาย* Colon (:) *ต่อท้ายด้วยเสมอเช่น*

LOOP : INPUT S*1*

IF S*1* = *1* THEN LOOP

2.4.*3* Variables *เป็นที่สำหรับเก็บข้อมูลชั่วคราว โดยจะต้องกำหนดขนาด (*Size) *ซึ่งอาจเป็น* bits, bytes *หรือ* Words *มีรูปแบบดังนี้*

Label VAR Size {.Modifier}

เช่น SW*1* VAR bit

หมายเหตุ : .Modifier *นั้น เป็น* Option *เพิ่มเติม สำหรับบอกว่า* Variable *ตัวนั้น สร้างมาได้อย่างไร*

2.4.4 Arrays Variable arrays *สามารถกำหนดชื่อตัวแปรหลายตัวในชื่อเดียวกันได้ มีรูปแบบ เช่น*

Label VAR Size [No. of elements]

หมายเหตุ: *จำนวน* element *สูงสุด มีได้ดังนี้*

Size	No. of elements
Bit	256
Byte	96 *
Word	48 *

* *จำนวน* element *ของ* byte *และ* word *ขึ้นตรงต่อขนาด* RAM Bank *ของ* MCU

2.4.5 Constants *เป็นชื่อที่กำหนดขึ้นแทนค่าคงที่ ซึ่งคล้ายกับกำหนดตัวแปร มีรูปแบบคือ*

Label CON Constant expression

2.4.6 String Constants *ให้ใช้เครื่องหมายกำหนดเช่นเดียวกับ* ASCII value *เช่น* Lcdout “Hello”

2.4.7 *การกำหนดค่า* PORT *และ* Register *ต่าง ๆ มีรูปแบบกำหนดดังต่อไปนี้คือ* PORT A = %01010101 ‘*ส่งข้อมูลออกที่* PORT A *ตามค่าที่กำหนด* VAR1 = PORT B & $0F’ *ทำ* Logic AND *ค่าที่* PORT B *ด้วยค่า* $0F *แล้วกำหนดเป็นค่า* VAR1

2.4.8 *การกำหนดค่า* Pins *ของ* MCU *มีรูปแบบกำหนดดังต่อไปนี้คือ* PORT B.1 = *กำหนดให้ ขา* 1 *ของ* PORT B *มีค่าเป็น* 1

2.4.9 *การกำหนดให้* PORT *ของ* MCU *เป็น* OUTPUT *หรือ* INPUT *สามารถกำหนดได้โดย การกำหนด* Tri-state Register (TRIS) *ของ* Port *เช่นเดียวกับการใช้ภาษา* Assembly *โดยถ้ากำหนด*

TRIS *มีค่าเป็น* 0 *จะทำให้* Port *นั้น เป็น* OUTPUT *ถ้ากำหนด* TRIS *มีค่าเป็น* 1 *จะทำให้* Port *นั้น เป็น* INPUT

2.4.10 *การเขียนคำสั่งหลายบรรทัด (*Multi-Statement Lines) *สามารถกำหนดได้ ดังนี้คือ*

W2 = W0 : W0 = L1 : W1 = W2

2.4.11 *การกำหนดบรรทัดคำสั่งต่อเนื่อง (*Line-extension) PIC BASIC *ได้กำหนดจำนวนตัวอักขระในการเขียนคำสั่งในแต่ละบรรทัดไว้ไม่เกิน* 250 *ตัว แต่ถ้าในบรรทัดคำสั่งเดียวกันนั้น เราต้องการแบ่งออกเป็น* 2 *บรรทัด โดยบรรทัดทั้ง* 2 *ยังคงต้องต่อเนื่องกัน สามารถใช้เครื่องหมาย (_) ใส่ไว้ท้ายบรรทัดแรก ก็จะทำให้* 2 *บรรทัดต่อเนื่องเป็นบรรทัดเดียวกัน เช่น* BRANCH B0, [Label1, Label 2, Label 3,…] , _Label 4, Labels]

2.4.12 INCLUDE *ใช้สำหรับรวมโปรแกรมต่อไฟล์กัน เข้าไปทำงานเป็นโปรแกรมเดียว กัน เช่น* INCLUDE “modedefs.bas”

2.4.13 DEFINE *ใช้กำหนดค่าพารามิเตอร์ของฮาร์ดแวร์ หรือ ซอฟท์แวร์ให้มีค่าและการทำงานที่ แตกต่างไปจากที่โปรแกรมได้กำหนดเป็นเบื้องต้นไว้เช่น* DEFINE QSC 4 ‘*กำหนด* OSC.Speed 4 MHz

2.4.14 *การคูณ (*Multiplication) PIC BASIC *ได้กำหนดการคูณไว้สองลักษณะคือ คูณ 16* bit *และคูณ 32 บิต เช่น* W*1* = W*0 * 1000* ‘*คูณค่าใน* W*0 กับ 1000 และเก็บผลลัพธ์เข้าไว้ใน* W*1* W*2* = W*0**1000* ‘*คูณค่าใน* W*0 กับ 1000 และเก็บ* high order *16* bits *ไว้ใน* W*2*

2.4.15 *การหาร (*Division) *เช่น* W*1* = W*0/1000* ‘*หารค่า* W*0 โดย 1000 และเก็บผลลัพธ์ไว้ใน* W*1 และ*W*2* = W*0//1000* ‘*หารค่า* W*0 โดย 1000 และเก็บเศษไว้ใน* W*2*

2.4.16 *การเลื่อนบิตข้อมูล (* Bit shifting) *ใช้เครื่องหมาย << และ >> สำหรับเลื่อน* bit *ข้อมูลไปทางซ้ายและขวา ตามลำดับ ค่าที่ใช้เลื่อนมีค่าระหว่าง 0 – 15 เช่น* B*0* = B*0 << 3* ‘Shift B*0 ไป 3 ตำแหน่ง และ*W*1* = W*0 >>1* ‘Shift W*0 ไป 1 ตำแหน่ง และนำผลลัพธ์ไปเก็บไว้ใน* W*1*

2.4.17 DCD *สำหรับหาค่า* decode *ค่า* bit number (*0-15) เป็นค่า* binary number *8* bit *เช่น* B*0* = DCD *2* ‘*กำหนด* B*0 = %00000100*

2.4.18 DIG *สำหรับกำหนดค่าจากจำนวนหลักของเลขฐาน 10 (ค่าที่กำหนด 0-4 คิดจากขวามือสุด) เช่น* B*0 = 123* ‘Set B*0* to *123 และ*B*1* = B*0* DIG *1* ‘*กำหนด* B*1 มีค่า 2 (หลักที่ 2 จากทางขวามือสุด)*

2.4.19 MAX and MIN *กำหนดค่าพิกัดสูงสุดและต่ำสุด* B*1* = B*0* MAX *100* ‘*ค่า* B*1 ระหว่าง 100 และ 255* B*1* = B*0* MIN *100* ‘*ค่า* B*1 ระหว่าง* B*0 กับไม่เกิน 100*

2.4.20 NCD *เปลี่ยนค่า* bit *ของ* Binary number *มาเป็นค่าจำนวนของ* Decimal (bit number *ค่า* 0-16) *เช่น* B0 = NCD % 01001000 ‘Set B0 *มีค่า* 7

2.4.21 REV *กลับค่าบิตของเลข* Binary *จำนวน* bit (0-16) *นับจากขวาสุด หรือ* LSB *เช่น* B0 = %10101100 REV 4 ‘Set B0 *เป็น* 10100011

2.4.21 Trigonometric SINE, COS *เช่น* B1 = COS B0, B2 = SIN B0

2.4.21 Square root, Absolute B0 = SQR W1, B1 = ABS B0

2.4.21 Bitwise Operators *การกระทำเกี่ยวกับบิต ซึ่งสามารถเลือกกระทำแยกบิต หรือ* Set *แต่ละบิต เช่น* B0 = B0 & % 00000001 ‘*แยกบิต* 0 *ของ* B0 *และ*B0 = B0 | % 00000001 ‘Set bit0 *ของ* B0

2.4.22 Comparison Operator *เป็นเครื่องหมายเปรียบเทียบปริมาณระหว่าง* expression *หนึ่ง กับอีก* Expression *หนึ่ง ได้แก่*

Operator	**Description**
= *หรือ* = =	*เท่ากัน*
< > *หรือ ! =*	*ไม่เท่ากัน*
<	*น้อยกว่า*
>	*มากกว่า*
< =	*น้อยกว่าหรือเท่ากับ*
> =	*มากกว่าหรือเท่ากับ*

เช่น IF i > *10* THEN loop

2.4.23 Logical Operators *เป็น* Operator *กระทำทางด้าน* Logic *โดยเปรียบเทียบว่าเป็นจริงหรือเท็จ ได้แก่*

Operator	Description
AND *หรือ 82*	Logic AND
OR *หรือ* ||	Logic OR
XOR *หรือ* ^^	Logic XOR
ANDNOT	Logic N Logic
ORNOT	NOR
XORNOT	Logic NXOR AND

เช่น IF (A = = big) AND (B > mean) THEN run

2.4.24 *คำสั่ง* ADCIN *เป็นคำสั่งสำหรับค่าสัญญาณอะนาล็อก สำหรับ* MCU *บางเบอร์ เช่น* PIC 16F874/877 *เป็นต้น*

รูปแบบ

```
ADCIN channel, Var
```

2.4.25 *คำสั่ง* BRANCH *เป็นคำสั่งสำหรับใช้กระโดดไปทำงานตามตำแหน่ง ตามค่าของตัวแปรที่เป็น* INDEX

รูปแบบ

```
BRANCH INDEX, [Label1, Label2, …]
```

2.4.26 *คำสั่ง* BUTTON *เป็นคำสั่งที่รออ่านค่าที่ขาของ* Port *แล้วเก็บค่าเข้าไว้ในตัวแปร โดยมีการแก้การเด้งของหน้าคอนแทค และ* auto-repeat *และทำให้* pin *นั้นเป็น* input *โดยอัตโนมัติ*

รูปแบบคำสั่ง

```
BUTTON Pin, Down, Delay, Rate, BVar, Action, Label
```

2.4.27 *คำสั่ง* CALL *เป็นคำสั่งสำหรับเรียกโปรแกรม* Assembly subroutine *มา* execute *เช่น* CALL pass.asm 'extention *ไม่ต้องใส่ก็ได้*

2.4.28 *คำสั่ง* CLEAR *เป็นคำสั่งกำหนดให้ค่าใน* RAM register *มีค่าเป็น* 0 *หรือ* clear *ตัวแปรที่กำหนดทั้งหมด*

2.4.29 *คำสั่ง* CLEARWDT *เป็นคำสั่ง* Clear Watchdog Timer

2.4.30 *คำสั่ง* COUNT *เป็นคำสั่งสำหรับนับจำนวนลูก* pulse *ที่ปรากฏที่* Pin *ในช่วงคาบเวลาที่กำหนด*

รูปแบบ

```
COUNT Pin, Period, Var
```

Pin *เป็นขาของ* MCU Port

Period *เป็นคาบเวลาที่รอการนับ*

Var *ตัวแปรที่กำหนดเพื่อเก็บค่าที่นับได้*

2.4.31 *คำสั่ง* INPUT *เป็นคำสั่งที่กำหนดให้ขา* Pin *ของ* Port *มีสถานะเป็น* Input

รูปแบบ

```
INPUT Pin
```

Pin *อาจเป็นค่าตัวแปร หรือ ขาของ* Port *โดยตรงก็ได้*

2.4.32 *คำสั่ง* HIGH *เป็นคำสั่งให้ขาของ* Port *มีสถานะลอจิก* "1" *เมื่อใช้คำสั่งนี้ที่ขา* Pin *ใด ขานั้นจะมีสถานะเป็น* OUTPUT *โดยอัตโนมัติ*

รูปแบบ

```
HIGH Pin
```

Pin *อาจเป็นชื่อขาของ* Port *หรือตัวแปรแทนขานั้น ๆ ก็ได้*

2.4.33 *คำสั่ง* LOW *เป็นคำสั่งให้ขาของ* Port *มีสถานะลอจิก* "1" *เมื่อใช้คำสั่งนี้ที่ขา* Pin *ใด ขานั้นจะมีสถานะเป็น* OUTPUT *โดยอัตโนมัติ*

รูปแบบ

```
LOW Pin
```

Pin *อาจเป็นชื่อขาของ* Port *หรือตัวแปรแทนขานั้น ๆ ก็ได้*

2.4.34 *คำสั่ง* PAUSE *เป็นคำสั่งหน่วงเวลามีหน่วยควบเป็นมิลิวินาที (*mS)

รูปแบบ

```
PAUSE Period
```

Period *มีหน่วยเป็น* mS *มีค่าอยู่ระหว่าง* 0 ~ 65535mS

2.4.35 *คำสั่ง* TOGGLE *เป็นคำสั่ง กลับสถานะของ* Pin *ในแต่ละ* Port *ให้มีลอจิกเป็นตรงกันข้ามกับของเดิม รูปแบบ*

```
TOGGLE Pin
```

2.4.36 *คำสั่ง* LCDOUUT *เป็นคำสั่งสำหรับส่งข้อความออกทีจอ* LCD *คำสั่งนี้* Support LCD Module *แบบ* 2 *บรรทัด หรือ* 1 *บรรทัดที่ใช้ตัว* Controller *ของ* Hitachi *เบอร์* 44780 *หรือเบอร์อื่นที่เข้ากันได้กับเบอร์ดังกล่าวนี้* LCD *โมดูลเหล่านี้ มีขั้วต่อ* 14-16 *ขั้วแบบ* Single row header

2.4.37 *คำสั่ง* FOR…NEXT *เป็นคำสั่งสำหรับสร้าง* Loop *การทำงานซ้ำแบบมีเงื่อนไขการจบ*

```
FOR ตัวแปร = ค่าเริ่มต้น TO ค่าสุดท้าย {STEP Val}

. . .

งานที่ทำซ้ำ
```

ตัวแปร เป็นค่าที่กำหนดจำนวนรอบที่ทำซ้ำ ค่าเริ่มต้น เป็นค่าที่เริ่มต้นนับทำซ้ำ ค่าสุดท้าย เป็นค่าที่กำหนดทำซ้ำถึงสุดท้าย STEP Val *เป็นค่าที่กำหนดให้ตัวแปรลดหรือเพิ่มค่าแต่ละครั้ง จากค่าเริ่มต้นจนถึงค่าสุดท้าย* NEXT *เป็นคำสั่งปิดท้ายเพื่อให้ย้อนกลับไปทำซ้ำ*

2.4.38 *คำสั่ง* IF. . THEN *คำสั่ง* IF. . THEN. . ELSE. .ENDIF *เป็นคำสั่งสำหรับตรวจสอบเงื่อนไข เพื่อกำหนดทิศทางการทำงานของโปรแกรมแบบมีทางเลือกมากกว่า 2 ทิศทาง รูปแบบ* IF *เงื่อนไขที่ตรวจสอบ* THEN *งานที่ทำเมื่อเงื่อนไขเป็นจริงที่ 1* ELSE *งานที่ทำเมื่อเงื่อนไขเป็นจริงที่ 2 หรือนอกเหนือจาก 1* ENDIF

2.4.39 *คำสั่ง* WHILE. . WEND *เป็นคำสั่งสำหรับทำซ้ำภายใต้เงื่อนไขที่ตรวจสอบเป็นจริง*

รูปแบบ

```
WHILE Conditon
Statement
    :
WEND
```

2.4.40 *คำสั่ง* GOSUB. . RETURN *เป็นคำสั่งให้โปรแกรมกระโดดไปทำงานที่* Subroutine *เสร็จแล้วกลับมายังโปรแกรมหลักด้วยคำสั่ง* RETURN

รูปแบบ

```
GOSUB ชื่อ Label
```

2.5 ซอฟแวร์ที่เกี่ยวข้องกับการใช้งานและการพัฒนาไมโครคอนโทรลเลอร์

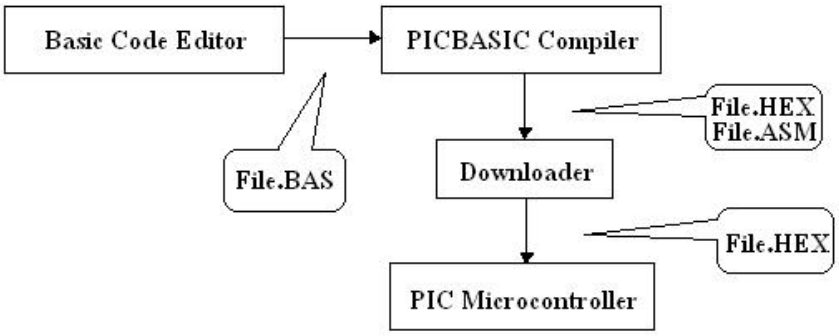
ภาษาระดับสูง PIC BASIC COMPILER *ประกอบด้วย 3 ส่วน คือ*

2.5.*1 ซอฟแวร์ที่ใช้สำหรับเขียนโปรแกรม ได้แก่* Code Designer *ในที่นี้ได้แก่ โปรแกรม* Microcode Studio Plus *ใช้สำหรับเขียนซอสโค้ดที่เป็นโปรแกรมภาษาเบสิก หรือเราเรียกว่า* BASIC Code Editor *ซึ่งไฟที่เกิดขึ้นจะมีนามสกุลเป็น* .bas (Mecanique,1970)

2.5.*2 ซอฟแวร์ที่ทำหน้าที่* Compile *ซอสโค้ดที่เป็นภาษาเบสิกไปเป็นภาษา* Assembly *คือ โปรแกรม* PBP246.ZIP (PIC Basic Pro ver 2.46) *ซึ่งไฟที่เกิดขึ้นจะมีนามสกุลเป็น* .asm

2.5.3 *โปรแกรม* mplab *จะทำหน้าที่แปลงภาษา* Assembly *เป็น* Machine code *โปรแกรมนี้เป็นของบริษัทไมโครชิพ ซึ่งไฟที่เกิดขึ้นจะมีนามสกุลเป็น* .hex

2.5.4 *โปรแกรม* EPICWIN *เป็นโปรแกรมที่ทำหน้าที่ดาวน์โหลดซอสโค้ด ที่เป็น* Machine code (File.hex) *ลงตัวชิพไมโครคอนโทรลเลอร์*



ภาพที่ 2-8 *ขั้นตอนในการพัฒนาไมโครคอนโทรลเลอร์ตระกูล* PIC

ซึ่งขั้นตอนการใช้งานโปรแกรม Microcode Studio Plus *อยู่ในภาคผนวก ก เรียบร้อยแล้ว*

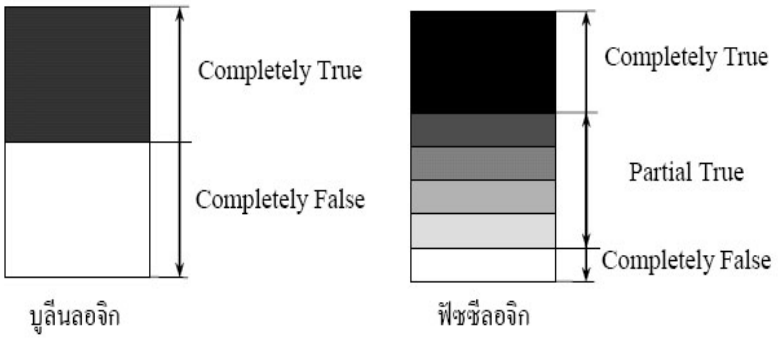
2.6 ตรรกศาสตร์คลุมเครือหรือระบบฟัซซี (Fuzzy)

ระบบฟัซซี เป็นระบบด้านคอมพิวเตอร์ที่ทำงานโดยอาศัยฟัซซีลอจิกที่คิดค้นโดย L. A. Zadeh *ในปี ค.ศ. 1965 ฟัซซีลอจิกเป็นตรรกะที่อยู่บนพื้นฐานความเป็นจริงที่ว่า"ทุกสิ่งบนโลกแห่งความเป็นจริงไม่ใช่มีเฉพาะสิ่งมีความแน่นอนเท่านั้น แต่มีหลายสิ่งหลายเหตุการณ์ที่เกิดขึ้นอย่างไม่เที่ยงและ ไม่แน่นอน (*Uncertain*) อาจเป็นสิ่งที่คลุมเครือ (*Fuzzy*) และไม่ชัดเจน (*Exact*) ส่วนฟัซซีลอจิกหรือตรรกะแบบคลุม เครือเป็นคณิตศาสตร์แขนงใหม่ที่มีความสำคัญต่อเทคโนโลยี*

สารสนเทศเป็นอย่างมาก ทั้งนี้เพื่อนำไปประยุกต์ใช้งานด้านต่างๆ ซึ่งนับวันยิ่งมีความต้องการระบบคอมพิวเตอร์ที่มีความสามารถในการปรับ เปลี่ยนระบบ ได้โดยอัตโนมัติ ตามสภาพแวดล้อมที่เปลี่ยนไปมีการตัดสินใจแบบชาญฉลาดเยี่ยงมนุษย์ได้มากขึ้น ซึ่งมนุษย์สามารถแก้ปัญหาต่าง ๆ ที่ไม่เคยพบได้โดยอาศัยความรู้เก่าที่ได้เรียนรู้มาประยุกต์ในการแก้ปัญหาได้อย่างมีประสิทธิภาพ ซึ่งในที่นี้ก็เพื่อนำมาประยุกต์ใช้งานด้านธุรกรรมการค้าขายระบบอิเล็กทรอนิกส์ในการหากลุ่มลูกค้าเป้าหมาย นำมาประกอบการสนับสนุนการตัดสินใจ ของผู้เชี่ยวชาญอย่างผู้บริหารได้ด้วยเช่นกัน

2.6.1 ฟัซซีเซต (Fuzzy Set) *เป็นเซตที่มีขอบเขตที่ราบเรียบ ทฤษฎีฟัซซีเซตจะครอบคลุมทฤษฎีเซตแบบฉบับ โดยฟัซซีเซตยอมให้มีค่าความเป็นสมาชิกของเซตระหว่าง 0 และ 1 ในโลกแห่งความเป็นจริง เซตไม่ใช่มีเฉพาะเซตแบบฉบับเท่านั้น จะมีเซตแบบฟัซซีด้วย ฟัซซีเซตจะมีขอบเขตแบบฟัซซีไม่ใช่เปลี่ยนแปลงทันทีทันใดจากขาวเป็นดำ ตัวอย่างเช่น เซตของลูกค้าที่มีการตัดสินใจซื้อสินค้าที่ตนต้องการ จะเห็นได้ว่าสมาชิกในเซตนี้จะเฉพาะบุคคลที่ต้องการระดับเดียวกันหมด แต่บางคนอาจจะมีความต้องการในสินค้าที่ต่างกันไป การใช้เซตแบบดั้งเดิมจึงไม่เหมาะสม (พยุง มีสัจ, 2552:40-78)*

*ตรรกะแบบฟัซซี (*Fuzzy Logic*) เป็นเครื่องมือที่ช่วยในการตัดสินใจภายใต้ความไม่แน่นอนของข้อมูล โดยยอมให้มีความยืดหยุ่นได้ ใช้หลักเหตุผลที่คล้ายการเลียนแบบวิธีความคิดที่ซับซ้อนของมนุษย์ ฟัซซีลอจิกมีลักษณะที่พิเศษกว่าตรรกะแบบจริงเท็จ (*Boolean logic*) เป็นแนวคิดที่มีการต่อขยายในส่วนของความจริง (*Partial True*) โดยค่าความจริงจะอยู่ในช่วงระหว่างจริง (*Completely True*) กับเท็จ (*Completely False*) ส่วนตรรกศาสตร์เดิมจะมีค่าเป็นจริงกับเท็จเท่านั้น แสดงดังภาพประกอบที่* 2-9



ภาพที่ 2-9 *ตรรกะแบบจริงเท็จ(บูลีนลอจิก) กับตรรกะแบบฟัซซี (ฟัซซีลอจิก)*

*(ก) ความเป็นฟัซซี (*Fuzziness*) มีชื่อเรียกว่า มัลติวาลานซ์ (*Multi valence*) ซึ่งมีค่าที่ความเป็นสมาชิกมากกว่า 2 ค่า และแตกต่างกับไบวาลานซ์ (*Bivalence*) ที่มีความเป็นสมาชิกเพียง 2 ค่า*

*(ข) ฟัซซีเซต (*Fuzzy set*) เป็นเครื่องมือทางคณิตศาสตร์ที่สื่อถึง "ความไม่แน่นอน (*Uncertainty*)" สามารถที่จะสร้างและกำหนดรูปแบบ (*Modeling*) ของลักษณะความ ไม่แน่นอนที่เป็นความคลุมเครือ ความไม่ตายตัว รวมถึงความขาดข้อมูลบางส่วน โดยทฤษฎีของฟัซซีเซตจะใช้ลักษณะความหมายตัวแปร (*Linguistic*) มากกว่าปริมาณ (*Quantitative*) ของตัวแปร*

*ภาพประกอบที่ 2-*10 *เป็นการแสดงให้เห็นว่าแนวทางในการตัดสินใจของปัญหาทั้งหมดมีเพียงส่วนน้อยที่เป็นสิ่งที่แน่นอน (*Certainty*) ที่เหลือคือสิ่งที่ไม่แน่นอนซึ่งประกอบด้วยความไม่แน่นอนที่มีลักษณะแบบสุ่ม และความไม่แน่นอนที่มีลักษณะเป็นฟัซซี หรือคลุมเครือ ซึ่งมีมากกว่าร้อยละ 40 เพราะปัญหาส่วนมากเกี่ยวข้องกับการตัดสินใจของมนุษย์ซึ่งจะตัดสินใจตามพื้นฐานความคิดของตนเป็นหลัก ถ้านำหลักการตัดสินใจในการจัดหากลุ่มลูกค้าเป้าหมาย จะก็สามารถจัดหาโดยผู้บริหาร ว่าบุคคลใดซื้อหรือไม่ซื้อในตัวสินค้าของแต่ละองค์กร ผู้ที่ไม่ได้ซื้ออาจจะเป็นลูกค้ากลุ่มเป้าหมายได้*

ภาพที่ 2-10 *ความไม่แน่นอน* (Uncertainty)

2.6.2 ความสัมพันธ์ทั่วไปของฟัซซี่ (Fuzzy Relations) จะแตกต่างจากความสัมพันธ์แบบดั้งเดิมที่มีเพียงความสัมพันธ์แบบขาว-ดำคือการมีค่าสมาชิก ตัวอย่างเช่น ความสัมพันธ์ไบนารี (Binary Relation) ของคำว่า "เพื่อน" จะถูกแบ่งออกเป็น 2 ความหมายคือ "เป็นเพื่อน" และ "ไม่เป็นเพื่อน" ซึ่งมีความแตกต่างกันอย่างชัดเจน ในความเป็นจริงแล้วเราสามารถใช้ n-ary relation ในการอธิบายความสัมพันธ์ระหว่างวัตถุ n ชิ้นได้เราสามารถใช้ n-ary relationในการอธิบายความสัมพันธ์

ระหว่างนักเรียน X ที่เรียนวิชา Y ในภาคการศึกษา Z ปีการศึกษา W ได้ ซึ่งจะเป็นความสัมพันธ์ที่เกิดจาก argument 4 ตัวเช่น tool_course (student, course, semeter, year) เราสามารถให้คำจำกัดความของ n-ary relation โดยกำหนดให้เป็นเซ็ตของรายการของวัตถุ n ชิ้น โดยที่แต่ละรายการจะอธิบายความสัมพันธ์ของมันเอง Binary relation ของตัวแปร x และ y ที่อยู่ใน X และ Y ถูกกำหนดให้เป็นคู่ลำดับใน $X \times Y$ ตัวอย่างเช่น binary relation ของ “less than” ระหว่างจำนวนจริง 1 คู่ กำหนดได้ดังนี้ $R = \{(x, y)\ x < y, x, y \in R\}$ มันเป็นเรื่องง่ายที่จะเห็นความสัมพันธ์ในรูปของ subset ของ $X \times Y$ โดยทั่วไป n-ary relation ของ x1, x2, …, xn จะมีโดเมนคือ X1, X2, …, Xn และมี subset คือ n *(Zdenko Kovaccic and Stjepan Bogdan, 2002)*

2.6.3 ฟัซซีพร็อพเพอตี้เซต (Fuzzy Property Set) *เป็นทฤษฎีที่ถูกนำเสนอโดย* Zdzislaw Pawlak *ในปีค.ศ.* 1982 *เป็นทฤษฎีคณิตศาสตร์แนวใหม่ ที่ใช้ในการจัดการเกี่ยวกับเรื่องความคลุมเครือและความไม่แน่นอนของข้อมูลซึ่งทฤษฎีนี้จะอิงพื้นฐานในเรื่องเกี่ยวกับปัญญา ประดิษฐ์ (*AI : Artificial Intelligence) *เป็นสำคัญและเป็นทฤษฎีที่สัมพันธ์และอยู่ในขอบเขตในแนว ทางเดียวกันกับการศึกษาเกี่ยวกับเรื่องการเรียนรู้, การวิเคราะห์การตัดสินใจ, การค้นหาความรู้ที่ต้องการจากฐานข้อมูล, ระบบผู้เชียวชาญ และอื่นๆ*

Fuzzy set *ถูกนำเสนอโดย* Lotfi A. Zadeh *ในปี* 1965 *โดยที่ฟัซซีเซตเป็นแนวทางในการแสดงถึงลักษณะความคลุมเครือไม่ชัดเจน (*Vagueness) *กำหนดขอบเขตในการแบ่งความเป็นสมาชิกและไม่เป็นสมาชิกของข้อมูลที่เราสนใจอย่างชัดเจนมากยิ่งขึ้นและจากคุณสมบัติของทั้งสองทฤษฎีที่กล่าวถึงข้างต้นทำให้สามารถนำมาสร้างเป็นหลักการและกระบวนการต่างๆ ได้ ดังขั้นตอนต่อไปนี้*

(ก) การนำตัวอย่างข้อมูลมาแยกคุณสมบัติ และ ผลลัพธ์ของคุณสมบัติ ของข้อมูลในรูปแบบของตาราง ซึ่งจะทำให้เราสามารถเห็นความคลุมเครือของข้อมูลได้ง่ายขึ้น

*(ข) จากลักษณะของข้อมูลที่มีความคลุมเครือและขัดแย้งกันของข้อมูล จะใช้ทฤษฎีของฟัซซีมาช่วยลดความคลุมเครือของข้อมูลซึ่งจะทำให้เราได้ผลลัพธ์ 2 อย่าง คือ ผลลัพธ์ที่หนึ่งเรียกว่า การประมาณขอบเขตล่าง (*Lower approximation) *ซึ่งจะได้กลุ่มตัวอย่างของข้อมูลที่ไม่คลุมเครือแน่นอนและผลลัพธ์ที่สองเรียกว่า การประมาณขอบเขตบน (*Upper approximation) *ซึ่งจะได้กลุ่มตัวอย่างข้อมูลที่มีความเป็นไปได้ว่าจะเป็นผลลัพธ์ที่แรก*

(ค) จะมีการนำตัวอย่างฐานข้อมูลจากขั้นตอนที่ 1 *โดยการนำค่าของข้อมูล (*Attribute) *ที่อยู่ในแต่ละคุณสมบัติมากำหนดเป็นค่าระดับความเป็นสมาชิก (*Degree of Membership) *ของผลลัพธ์ตามค่าข้อมูลที่มีอยู่ในตาราง โดยอาศัยหลักการแบบจำลองคุณสมบัติแบบฟัซซีเซต (*Fuzzy Property set Model) *เพื่อให้เกิดการแบ่งขอบเขตของความเป็นสมาชิกและไม่เป็นสมาชิกของคุณสมบัติในแต่ละผลลัพธ์*

(ง) นำข้อมูลต่างๆ จากข้อสามที่มีการกำหนดค่าระดับความเป็นสมาชิกเรียบร้อยแล้ว ทดลองโดยใช้หลักการผสมผสานระหว่างทฤษฎีของคริสเซตและฟัซซีเซตเพื่อทำการเปรียบเทียบกับผลลัพธ์ในขั้นตอนที่หนึ่งโดยการใช้สมการการวัดคุณภาพ ของการประมาณขอบเขตบน และสมการวัดคุณภาพของการประมาณขอบเขตล่าง โดยสามารถหาได้จากการนำจำนวนสมาชิกในขอบเขตล่างหารด้วยจำนวนตัวอย่างข้อมูลทั้งหมด และในส่วนคุณภาพของการประมาณขอบเขตบนก็สามารถคำนวณได้ในทำนองเดียวกันคือนำสมาชิกในขอบเขตบนหารด้วยจำนวนตัวอย่างข้อมูลทั้งหมด

2.7 ผลงานวิจัยที่เกี่ยวข้องและเอกสารอ้างอิง

งานวิจัยของ Hoa Ying เป็นการนำเสนอการควบคุมตำแหน่งของมอเตอร์ไฟฟ้ากระแสตรงขนาดเล็กด้วยฟัซซี ซึ่งผลของการทดสอบสมรรถนะของการควบคุมถูกนำมาเปรียบเทียบกับการควบคุมด้วยตัวควบคุมแบบ PD ผลการวิเคราะห์ข้อมูลที่ได้ให้ผลดีโดยเฉพาะสามารถชดเชยช่วงที่ไม่เป็นเชิงเส้นของกระบวนการได้ดี ซึ่งการทดลองนี้ไม่ได้ใช้ไมโครคอนโทรเลอร์เข้ามาเกี่ยวข้องเลย และการสร้างกฎเพื่อใช้ในการตัดสินใจไม่ครอบคลุมเท่าที่ควร *(Hoa Ying, 2000)*

งานวิจัยของ Hoang Le-Huy and Maher Hamdi ได้อธิบายถึงการควบคุมมอเตอร์กระแสตรงด้วยฟัซซีลอจิกกับแบบดังเดิมที่ไม่สามารถลดค่าความผิดพลาดของตำแหน่ง ณ สภาวะคงตัวลงได้และไม่ทำให้ผลการตอบสนองแบบชั่วครู่ของระบบเปลี่ยนไปและสร้างกฎพื้นฐานทางฟัซซีเพิ่มเติมอีกสองกฎ จากผลการทดลองพบว่าตัวควบคุมแบบฟัซซีที่ปรับปรุงแล้วสามารถลดค่าความผิดพลาดของตำแหน่ง ณ สภาวะคงตัว ลงได้มากกว่า 80 เปอร์เซ็นต์ เมื่อเทียบกับตัวควบคุมฟัซซีลอจิกแบบดั้งเดิม ซึ่งงานวิจัยนี้ได้พัฒนาเพิ่ม เติมจาก Hoa Ying แต่ยังไม่ได้นำมาใช้เพื่อการใช้งานกับหุ่นยนต์เพราะว่าต้องมีตัวแปรหลายตัวที่เกี่ยวข้อง เพื่อทำให้การทำงานของหุ่นยนต์มีประสิทธิภาพ *(Hoang Le-Huy and Maher Hamdi, 1993)*

งานวิจัยของ รัฐพงศ์ ปฏิกานัง เป็นการนำเสนองานวิจัยแบบจำลองทางคณิตศาสตร์ของระบบกันสะเทือนแบบบังคับของยานพาหนะที่มีองศาอิสระ งานวิจัยนี้เน้นศึกษาการยึดเกาะถนนที่ดีและมีระบบกันสะเทือนของการขับขี่ยานพาหนะ จากผลการคำนวณพบว่าระบบกันสะเทือนแบบบังคับที่ใช้การควบคุมแบบฟัซซีที่มีการปรับค่าฟัซซีของแรงที่ควบคุมการสั่นด้วยวิธี GA (Genetic Algorithm) จะมีสมรรถนะโดย รวมดีกว่าระบบกันสะเทือนที่ไม่มีตัวควบคุม ที่มีตัวควบคุมแบบ PID และที่มีตัวควบคุมแบบฟัซซีอย่างเดียว ซึ่งงานวิจัยนี้ยังขาดในเรื่องประสิทธิภาพการเคลื่อนที่ในสภาวะมีสิ่งกีดขวางและเส้นทางคดเคี้ยว *(รัฐพงศ์ ปฏิกานัง , 2548)*

งานวิจัยของ สุชิน มุขศรี เป็นการศึกษา การเรียนรู้การนำทางของหุ่นยนต์ของหุ่นยนต์จะใช้อุปกรณ์ตรวจจับวัตถุอ่านค่าระยะของวัตถุซึ่งใช้ในการอ้างอิงตำแหน่งในการเคลื่อนที่ ข้อมูลทั้งหมดจะถูกเก็บลงในฐานข้อมูลเพื่อใช้ในการควบคุมให้หุ่นยนต์เคลื่อนที่ตามเส้นทางด้วยตัวเอง แต่การทดลองนี้เน้นการเคลื่อนที่ของหุ่นยนต์ตามเส้นทางที่กำหนดเท่านั้น ไม่ได้นำเทคนิคอื่นใดเข้ามาปรับปรุงประสิทธิ ภาพการเคลื่อนที่ *(สุชิน มุขศรี, 2550)*

งานวิจัยของ Satean Tunyasrirut and Santi Wangnipparnto เป็นการศึกษาการควบคุมตำแหน่งของตัวกระตุ้นนิวแมติกเชิงเส้นโดยใช้ฟัซซีลอจิก ซึ่งการควบคุมตำแหน่งพิจารณาจากตารางการเคลื่อนที่บนแกนเดี่ยว ตัวควบคุมการทำงานหลักคือไมโครคอนโทรลเลอร์ MCS-51 โปรแกรมด้วยภาษา C-51 การดำเนินการของระบบสามารถปรับปรุงการเคลื่อนที่ของกระบอกสูบนิวแมติกตามค่าของข้อมูลนำเข้าต่างๆ ได้เป็นอย่างดี โดยใช้ฟัซซีลอจิกเข้ามาประยุกต์ใช้งานเพื่อกำหนดเป็นฟัซซีเซตและค่าความเป็นสมาชิกตามข้อมูลนำเข้าแล้วสร้างเป็นกฎขึ้นมาให้ครอบคลุมการทำงานของระบบ ส่งผลให้การควบคุมการทำงานของระบบดีกว่าเดิมเมื่อนำฟัซซีลอจิกมาใช้ แต่งานวิจัยนี้ยังขาดในเรื่องประสิทธิภาพการเคลื่อนที่ในสภาวะมีสิ่งกีดขวางและความยืดหยุ่นต่อการใช้งานที่ต้องปรับเปลี่ยนตามสภาพแวดล้อม *(Satean Tunyasrirut and Santi Wangnipparnto, 2006)*