บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

หุ่นยนต์เดินสองขาแบบ 5 องศาอิสระ นั้นเกิดจากความต้องการที่จะสร้างให้ links ต่างๆที่ต่อกันได้เกิดการเคลื่อนไหว และมีการเคลื่อนตำแหน่ง ไปในทิศทางที่กำหนดได้ ในการจัดทำหุ่นยนต์เดิน 2 ขาแบบ 5 องศา-อิสระ จำเป็นต้องอาศัยทฤษฎีและเอกสารที่เกี่ยวข้องต่างๆเพื่อนำมาสร้างหุ่นยนต์ดังนี้

2.1 ไมโครคอนโทรลเลอร์ MCS-51

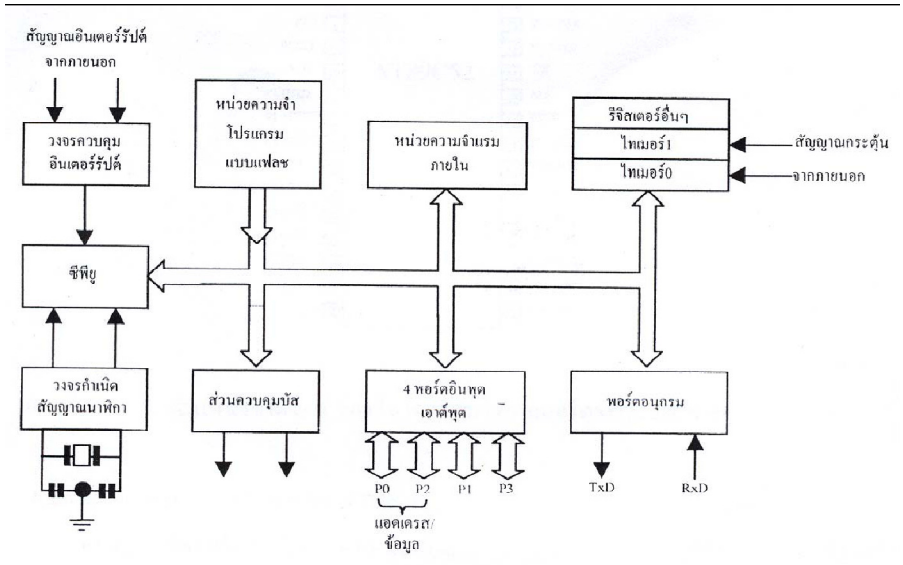
2.2 อาร์.ซี.เซอร์โวมอเตอร์

2.3 จลนศาสตร์

2.4 จุดเซ็นทรอยด์ของเส้น (Centroid of line)

2.1 ไมโครคอนโทรลเลอร์ MCS-51

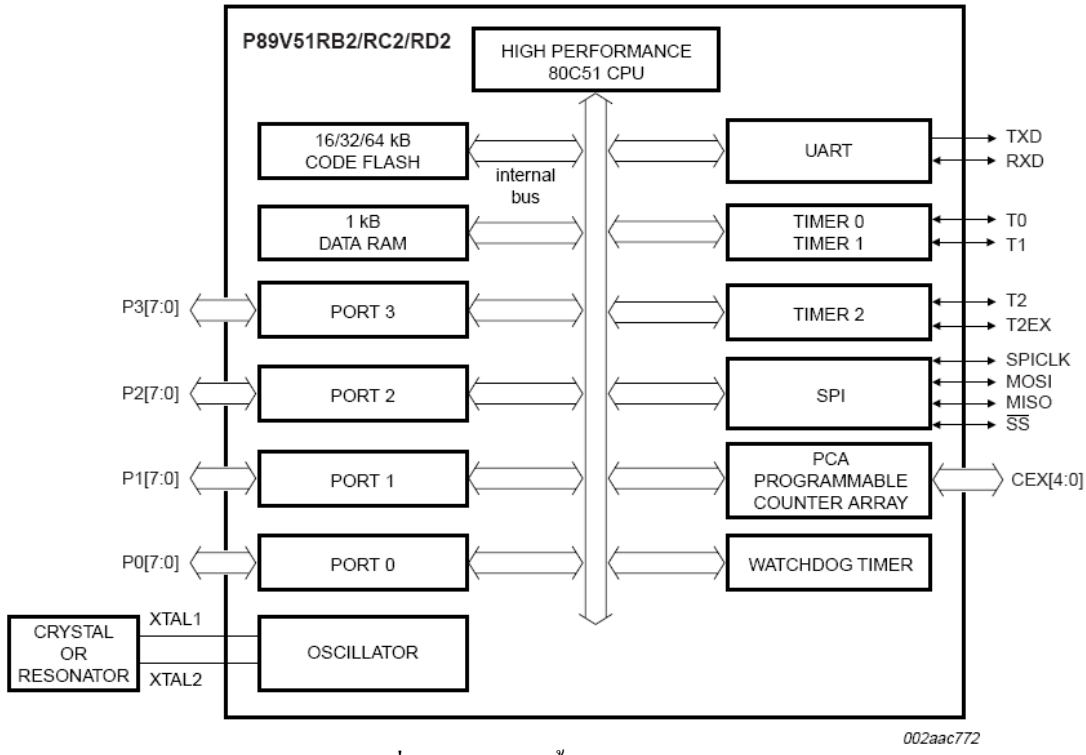
ไมโครคอนโทรลเลอร์นั้น เปรียบเสมือนเครื่องคอมพิวเตอร์ขนาดเล็ก ที่สร้างขึ้นมาให้อยู่ในไอซีเพียงตัวเดียว นั้นคือ โครงสร้างภายในของไมโครคอนโทรลเลอร์นั้นจะประกอบไปด้วย ระบบประมวลผลกลาง (CPU) , ระบบหน่วยความจำ (Memory system) และระบบอินพุตเอาท์พุต (I/O) อยู่ในตัวเองทั้งหมดนอกจากนี้ ไมโครคอนโทรลเลอร์บางรุ่น ยังได้เพิ่มวงจรเกี่ยวกับการสื่อสารข้อมูลแบบอนุกรม (Serial Communication) และวงจรที่ทำหน้าที่ทางด้านการแปลงสัญญาณระหว่างแบบอนาล็อกและดิจิตอล (A/D Convertor) เอาไว้ในตัวอีกด้วย ดังในภาพที่2.1 จึงทำให้มีความนิยมใช้ไมโครคอนโทรลเลอร์ ไปควบคุมการทำงานของอุปกรณ์ทางด้านอิเล็กทรอนิกส์ และคอมพิวเตอร์



ภาพที่2.1 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ตระกูล MCS-51

2.1.1 คุณสมบัติของ MCS-51

MCS-51 นั้นเป็นไมโครคอนโทรลเลอร์ตระกูลหนึ่งที่ได้รับความนิยมมาช้านาน และได้รับการตอบรับเป็นอย่างดีจากผู้ผลิตไอซี ซึ่งสังเกตได้จากการที่โครงสร้างหลักของไมโครคอนโทรลเลอร์นี้ยังเหมือนเดิม โค้ดโปรแกรมเดิมนั้น ยังคงนำมาใช้ได้กับไมโครคอนโทรลเลอร์รุ่นใหม่ๆ ที่ออกมาสู่ท้องตลาด ทั้งที่ไมโครคอนโทรลเลอร์รุ่นใหม่ๆที่ออกมานั้นมีประสิทธิภาพ และความเร็วสูงกว่าเดิมมากขึ้นเรื่อยๆ เช่นคุณสมบัติของไมโครคอนโทรลเลอร์ที่เลือกใช้คือ P89V81RD2BN ดังในภาพที่2.2 ดังนี้



ภาพที่2.2 โครงสร้างพื้นฐาน P89V51XX

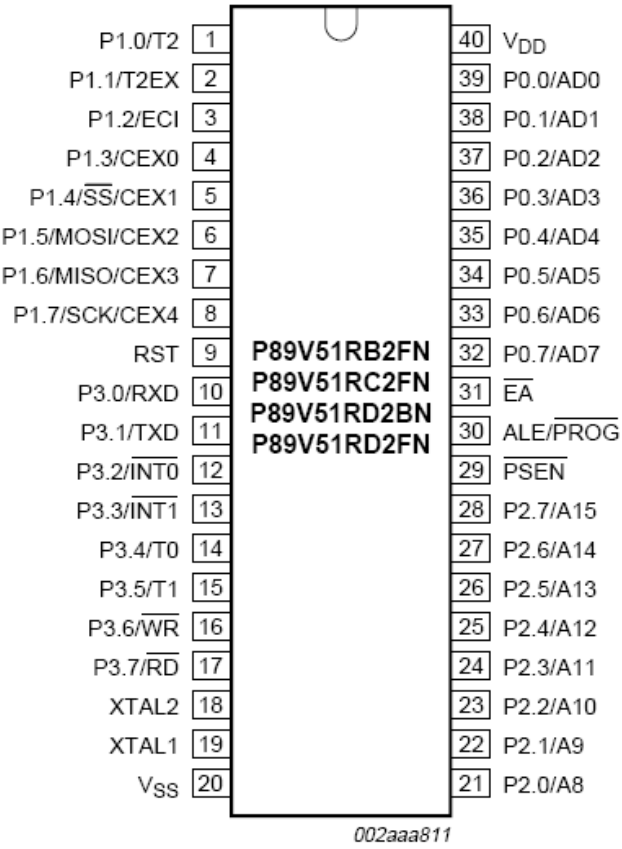
คุณสมบัติที่สำคัญ ๆ ของชิปไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีดังนี้

- ใช้ตัวประมวลผลกลาง 80C51
- ต้องการแหล่งจ่ายไฟ 5 โวลต์ เพียงชุดเดียว
- มีหน่วยความจำสำหรับเก็บโปรแกรมควบคุมการทำงานอยู่ภายในชิป จำนวน 64 กิโลไบต์
- มีหน่วยความจำสำหรับเก็บข้อมูลชั่วคราว (RAM) อยู่ภายในชิปจำนวน 1024 ไบต์
- สามารถใช้หน่วยความจำสำหรับโปรแกรมและข้อมูลที่อยู่ภายนอกชิปได้อย่างละ 64 กิโลไบต์ แยกจากกัน
- คำสั่งส่วนใหญ่ใช้เวลาทำงานเพียง 1 ไมโครวินาที เมื่อใช้คริสตอลความถี่ 12 เมกะเฮิรตซ์
- มีพอร์ตที่สามารถรับหรือส่งข้อมูลได้ทั้ง 2 ทิศทาง จำนวน 4 พอร์ต ๆ ละ 8 บิต หรือสามารถใช้งาน เป็นพอร์ต ขนาด 1 บิตแยกจากกันทำให้เสมือนมี พอร์ตขนาด 1 บิต ใช้งานรวมทั้งสิ้น 32 พอร์ต
- รับและส่งข้อมูลแบบอนุกรมได้ในตัว

- จัดลำดับความสำคัญของสัญญาณอินเตอร์รัปต์ได้ 8 ระดับ
- มีจิสเตอร์สำหรับใช้งานเป็นไทม์เมอร์หรือเคาน์เตอร์ เพื่อนับจำนวนสัญญาณนาฬิกาภายในชิป หรือนับการเปลี่ยนสถานะของสัญญาณภายนอกขนาด 16 บิต จำนวน 2 ตัว เพื่อใช้สำหรับนับจำนวนพัลส์ วัดความกว้างของพัลส์หรือ ใช้วัดช่วงเวลา
- ความจำสำหรับเก็บข้อมูลภายในบางส่วน สามารถเข้าถึงข้อมูลได้ทั้งระดับไบต์, ระดับบิต เพื่อให้การออกแบบโปรแกรมและการควบคุมระบบทำได้ง่ายขึ้น
- มีคำสั่งคูณและหารเลขขนาด 8 บิตในตัวเอง
- สามารถประมวลผลแบบบูลีนเพื่อใช้ในงานควบคุมโดยเฉพาะโครงสร้างของไมโครคอนโทรลเลอร์ตระกูลMCS-51

2.1.2 ตำแหน่งขาของไมโครคอนโทรลเลอร์ P89V51XX

ไมโครคอนโทรลเลอร์จะมีตำแหน่งขาดังแสดงใน ภาพที่2.3



ภาพที่2.3 แสดงตำแหน่งขาของไมโครคอนโทรลเลอร์

หน้าที่การใช้งานแต่ละขาของชิปไมโครคอนโทรลเลอร์ P898V51XX มีดังนี้

2.1.2.1 ขา V_{SS} (ขา 20) สำหรับต่อลงกราวด์

2.1.2.2 ขา V_{CC} (ขา 40) สำหรับต่อแหล่งจ่ายแรงดันกระแสตรงขนาด 5 โวลต์

2.1.2.3 ขาพอร์ต 0 (ขา 32 - 39) มี 8 ขา ใช้เป็นขาสำหรับพอร์ต 0 ขนาด 8 บิต (PO.O-PO.7) แบบ Open Drain Bidirectional พอร์ตนี้สามารถใช้งานเป็น อินพุตเอาต์พุตพอร์ตทั่วไปได้ โดยหากใช้งานเป็นอินพุตพอร์ต ต้องโหลดค่า 1 ไปยัง แต่ละบิตของ พอร์ตนี้เพื่อ บังคับให้ ขาอยู่ในสถานะถูกปล่อยลอย (มี สถานะ high impedance) นอกจากใช้งานเป็นอินพุต เอาต์พุต พอร์ตแล้ว พอร์ต 0 ยังใช้ในการติดต่อหน่วยความจำสำหรับเก็บโปรแกรม และ ข้อมูล - ภายนอกชิปด้วย โดยส่งค่าแอดเดรสบัสไบต์ต่ำ (AO-A7) จากหน่วยความจำ ภายนอก ในระหว่างการเขียนหรืออ่านข้อมูล โดยมีวงจร พูลอัพภายใน

2.1.2.4 ขาพอร์ต 1 (ขา 1 – 8) มี 8 ขา ใช้เป็นขาสำหรับพอร์ต 1 (P1.0 – P1.7) สามารถใช้งานเป็นอินพุตหรือเอาต์พุตพอร์ตทั่วไปได้ หากต้องการใช้งานเป็น อินพุตพอร์ต ต้องโหลด ค่า 1 ไปยังแต่ละบิตของพอร์ตนี้ เพื่อให้มีสถานะ high impedanceโดยมีวงจรพูลอัพภายใน

2.1.2.5 ขาพอร์ต 2 (ขา 21 – 28) มี 8 ขา ใช้เป็นขาสำหรับพอร์ต 2 (P2.0 – P2.7) ขนาด 8 บิต แบบ Open Drain Bidirectional พอร์ตนี้สามารถใช้งานเป็น อินพุตเอาต์พุตพอร์ตทั่วไปได้ โดยหากใช้งานเป็นอินพุตพอร์ต ต้องโหลดค่า 1 ไปยังแต่ละบิตของพอร์ตนี้ เพื่อบังคับให้ขาอยู่ในสถานะ high impedance นอกจากจะใช้งานเป็นอินพุตเอาต์พุตพอร์ตทั่วไปแล้ว พอร์ต 2 ยังใช้ในการ ติดต่อหน่วยความจำสำหรับเก็บโปรแกรมและข้อมูลภายนอกด้วย โดยใช้ สำหรับค่าส่งแอดเดรสบัสไบต์สูง (A8 – A15) และมีวงจรพูลอัพภายใน

2.1.2.6 ขาพอร์ต 3 (ขา 10 – 17) มี 8 ขา ใช้เป็นขาสำหรับพอร์ต 3 (P3.0 – P3.7) สามารถ ใช้งานเป็นอินพุตเอาต์พุตพอร์ตทั่วไปได้ หากต้องการใช้ งานเป็นอินพุตพอร์ต ต้องโหลดค่า 1 ไปยังแต่ละบิตของพอร์ตนี้ เพื่อให้มี สถานะ high impedance โดยใช้วงจรพูลอัพ ภายใน นอกจากนี้ยังใช้งานใน หน้าที่พิเศษต่าง ๆ อีกหลายอย่างดังนี้

- ขา P3.0 ใช้รับข้อมูลจากภายนอกแบบอนุกรม
- ขา P3.1 ใช้ส่งข้อมูลออกไปภายนอกแบบอนุกรม
- ขา P3.2 ใช้เป็นอินพุตเพื่อรับสัญญาณอินเตอร์รัปต์ชนิดที่ 0
- ขา P3.3 ใช้เป็นอินพุตเพื่อรับสัญญาณอินเตอร์รัปต์ชนิดที่ 1
- ขา P3.4 สัญญาณอินพุตให้เคาน์เตอร์ของไทม์เมอร์ 0
- ขา P3.5 สัญญาณอินพุตให้เคาน์เตอร์ของไทม์เมอร์ 1
- ขา P3.6 ใช้เป็นสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำ สำหรับเก็บข้อมูลภายนอกชิป
- ขา P3.7 ใช้เป็นสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำสำหรับเก็บข้อมูลภายนอกชิป
- การใช้งานพอร์ต 3 ในหน้าที่พิเศษดังกว่าวนี้จะต้องโหลดค่า 1ไปยังแต่ละบิตที่ต้องการใช้ก่อนทุกครั้ง

2.1.2.7 ขา RST (ขา 9) ใช้สำหรับการรีเซตวงจรทุกอย่างภายในชิป เพื่อเริ่มต้นการ ทำงานใหม่ การรีเซตใช้เมื่อเริ่มจ่ายพลังงานหรือเมื่อโปรแกรมเกิดทำงาน ผิดพลาด เมื่อต้องการรีเซตชิป MCS-51 ขานี้ต้องมีสถานะ 1 เป็นเวลาอย่าง น้อย 2 แมชชีนไซเกิลระหว่างที่ออสซิลเลเตอร์ยังทำงานอยู่ โดยต้องต่อตัวต้านทานค่า 8.2 กิโลโอห์มเพื่อทำหน้าที่พูลดาวน์ (รักษาค่าแรงดันไฟฟ้าให้มี สถานะเป็นกราวด์) และเพื่อให้ตัวชิปรีเซตเองเมื่อเริ่มจ่ายพลังงานให้ต่อตัวเก็บประจุขนาด 10 ไมโครฟารัดคร่อมระหว่างขา RST กับ Vcc

2.1.2.8 ขา ALE/$\overline{PROG}$ (ขา 30) เป็นขาสำหรับใช้ส่งสัญญาณออกไปภายนอก เพื่อ ควบคุมการแลตช์ค่าแอดเดรสไบต์ต่ำ (address latch enable) จากพอร์ต 0 ในระหว่างการติดต่อหน่วยความจำสำหรับเก็บโปรแกรมหรือข้อมูลภายนอก ปกติเมื่อไม่มีการติดต่อหน่วยความจำภายนอกขานี้ จะส่งสัญญาณพัลส์ออกมา ด้วยความถี่ 1/8 ของความถี่ออสซิลเลเตอร์ที่ใช้ตลอดเวลา ดังนั้นเราสามารถ ใช้ความถี่ที่ได้จากขานี้ไปใช้งานอย่างอื่นได้ แต่ความถี่ที่ขานี้จะลดลงครึ่งหนึ่ง ในระหว่างติดต่อกับหน่วยความจำสำหรับเก็บข้อมูลที่อยู่ภายนอกชิป

2.1.2.9 ขา $\overline{PSEN}$ (ขา 29) ใช้ส่งสัญญาณสโตรบเพื่ออ่านคำสั่งจากโปรแกรมที่เก็บไว้ ในหน่วยความจำภายนอกชิป (program strobe enable) เมื่อชิปทำงานด้วย โปรแกรมที่เก็บไว้ในหน่วยความจำภายนอกขานี้ จะส่งสัญญาณสโตรบสองครั้ง ในแต่ละแมชชีนไซเกิล แต่ในช่วงการเขียนหรืออ่านข้อมูลกับหน่วยความจำ ภายนอก หรือเมื่อใช้โปรแกรมจากหน่วยความจำสำหรับเก็บโปรแกรมภายใน ชิปจะไม่มีสัญญาณออกมาจากขานี้

2.1.2.10 ขา $\overline{EA}$ (ขา 31) เป็นขาสำหรับใช้เลือกให้ MCS-51 ทำงานจาก โปรแกรมที่อยู่ภายในหรือภายนอกชิป โดยหากขานี้มีสถานะเป็น 0 หมายถึงให้ใช้โปรแกรมจากหน่วยความจำที่เก็บโปรแกรมภายนอก หากขานี้มีสถานะ เป็น 1 หมายถึงบังคับให้ MCS-51 ใช้โปรแกรมจากหน่วยความจำสำหรับ เก็บโปรแกรมภายในชิป และสำหรับ MCS-51 ที่มีหน่วยความจำสำหรับเก็บ โปรแกรมภายในชิป สามารถเลือกให้ทำงานได้ทั้งจากโปรแกรมที่เก็บใน หน่วยความจำภายในชิป หรือจากโปรแกรมที่เก็บไว้ในหน่วยความจำภายนอก ชิปด้วยการต่อขา EA กับไฟเลี้ยงหรือกราวด์ตามลำดับ ส่วนใน MCS-51 ที่ ไม่มีหน่วยความจำสำหรับเก็บโปรแกรมภายในชิป ให้ต่อขานี้ลงกราวด์เสมอ

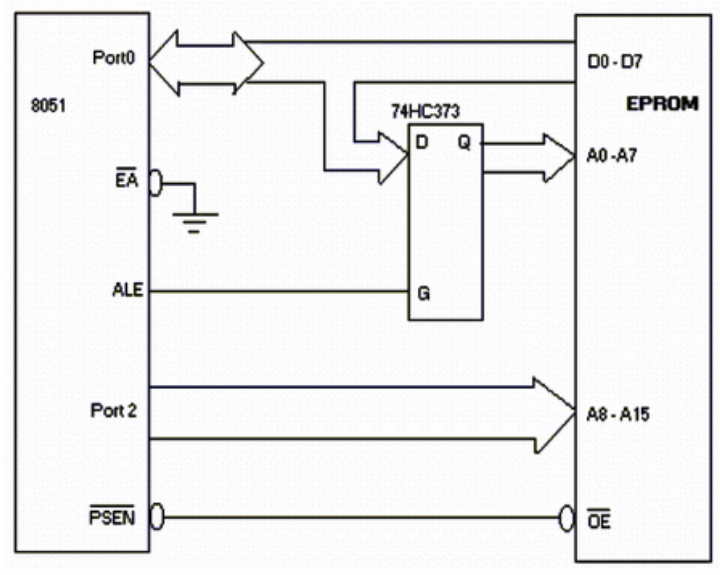
2.1.2.11 ขา XTAL 1 (ขา 19) ใช้ต่อคริสตอลภายนอก โดยเป็นอินพุตเข้าสู่วงจรออสซิลเลเตอร์

2.1.2.12 ขา XTAL 2 (ขา 18) ใช้ต่อคริสตอลภายนอก โดยเป็นเอาต์พุตออกจากวงจรออสซิลเลเตอร์

2.1.3 หน่วยความจำ

ไมโครคอนโทรลเลอร์จำมีความจำขนาด 64 กิโลไบต์ มีหน่วยความจำที่เก็บโปรแกรมได้ 8 กิโลไบต์ ในการใช้งานเราสามารถเก็บโปรแกรมเข้าในตัวไมโครคอนโทรลเลอร์ได้

ในกรณีที่ผู้ออกแบบต้องการใช้หน่วยความจำภายนอก ไม่ว่าจะเป็นหน่วยความจำสำหรับเก็บข้อมูลหรือสำหรับโปรแกรม พอร์ต 0 จะถูกกำหนดการใช้งานเป็นดาต้าบัสและแอดเดรสบัสไบต์ต่ำ ส่วนพอร์ต 2 จะถูกกำหนดการใช้งานเป็นตัวส่งค่าแอดเดรสบัสไบต์สูง



ภาพที่2.4 การเชื่อมต่อหน่วยความจำข้อมูลภายนอกของไมโครคอนโทรลเลอร์ที่ใช้ CPU 8051

บางส่วนของพอร์ต 3 จะถูกใช้ส่งสัญญาณควบคุมหรือคอนโทรลบัส แต่หากหน่วยความจำที่ใช้ภายนอกต้องการไม่ถึง 64 กิโลไบต์ พอร์ต 2 ที่ใช้เป็นแอดเดรสบัสไบต์สูงจะไม่ถูกนำมาใช้ทั้งหมด แต่พอร์ต 0 จะถูกใช้ทั้งหมด 8 เส้น เพราะต้องใช้เป็นดาต้าบัส ส่วนพอร์ต 3 จะนำมาใช้ติดต่อกับหน่วยความจำด้วยหรือไม่นั้นขึ้นอยู่กับหน่วยความจำที่ใช้ภายนอกว่า มีหน่วยความจำส่วนที่ใช้เก็บข้อมูลด้วย หรือไม่ ดังนั้นในการออกแบบระบบ หากต้องการใช้หน่วยความจำภายนอกมากขึ้นเพียงใด ก็จะยิ่งทำให้เหลือจำนวนพอร์ตที่จะนำมาใช้งานลดลง ในการออกแบบจริงจึงต้องพยายามลดขนาดหน่วยความจำภายนอกให้เหลือน้อยที่สุด

2.1.4 ไทม์เมอร์ / เคาร์เตอร์ (Timer/Counter)

ไทม์เมอร์ / เคาร์เตอร์ คือวงจรที่ทำหน้าที่นับความถี่ของสัญญาณนาฬิกา หรือนับจำนวนครั้งความเปลี่ยนแปลงของลอจิก เราสามารถนำความสามารถของระบบ ไทม์เมอร์ / เคาร์เตอร์ ไปประยุกต์ใช้ได้หลายอย่าง ไม่ว่าจะนำไปใช้ทำเป็น RTC (Real Timer Clock) ของระบบงานที่เราต้องการ, เอาไว้สร้างสัญญาณพัลซ์ (Pulse) หรือทำระบบนับจำนวนก็ได้ โดยหลักการทำงาน ไทม์เมอร์ / เคาร์เตอร์ ก็คือการนับจากค่าที่ตั้งเอาไว้ หรือจากค่า 0 ไปยังจำนวนสูงสุดที่นับได้ เช่น $FFFF_{16}$ หรือ FF_{16} เมื่อถึงค่าสูงสุดก็จะเกิดการล้น (Overflow) โดยดูได้จากบิตตรวจการล้นของไทม์เมอร์ / เคาร์เตอร์ (TF0 สำหรับไทม์เมอร์ / เคาร์เตอร์ 0, TF1 สำหรับไทม์เมอร์ / เคาร์เตอร์ 1) เมื่อเกิดการล้น ไมโครคอนโทรลเลอร์ก็จะสร้างสัญญาณอินเทอร์รัพท์ของ ไทม์เมอร์ / เคาร์เตอร์ ตัวนั้นขึ้นมา เสร็จแล้วไมโครคอนโทรลเลอร์จะทำการเปลี่ยนค่าจากค่าสูงสุด ($FFFF_{16}$ หรือ FF_{16}) ให้กลับเป็น 0 อีกครั้ง ความแตกต่างในการทำงานระหว่าง ไทม์เมอร์ กับเคาน์เตอร์ ก็คือ

ไทม์เมอร์ เป็นการนับจำนวนสัญญาณนาฬิกาของตัวไมโครคอนโทรลเลอร์ โดยจะนับจำนวนสัญญาณนาฬิกา จำนวน 12 ลูกต่อการเปลี่ยนแปลง 1 ครั้ง นั่นหมายความว่า ไทม์เมอร์ จะนับค่าเพิ่มขึ้น 1 ค่าเมื่อเกิดสัญญาณนาฬิกาไปแล้วเป็นจำนวน 12 ลูก ดังนั้น ถ้าเราใช้คริสตอลความถี่ 12MHz สำหรับสร้างสัญญาณนาฬิกาให้ไมโครคอนโทรลเลอร์เราก็จะสามารถใช้ ไทม์เมอร์ ที่มีความถี่สูงสุด 1 MHz

เคาน์เตอร์ จะเป็นการนับสัญญาณจากภายนอก โดยใช้ขา T0/T1 ในการรับข้อมูล การนับของ Counter 1 ครั้ง จะต้องใช้เวลาประมาณ 2 machine cycle หรือสัญญาณนาฬิกาจำนวน 24 ลูก ทำให้เมื่อเราใช้คริสตอลความถี่ 12 MHz เราจะสามารถนับความถี่ได้สูงสุด หรือไม่เกิน 500 KHz (500,000 ครั้งใน 1 วินาที) นั้นเอง

2.1.4.1 รีจิสเตอร์ที่ใช้ร่วมกับ ไทม์เมอร์ / เคาน์เตอร์

รีจิสเตอร์ที่ใช้กับตัวตั้งเวลาจะเป็นรีจิสเตอร์ SFR มี 4 ชุดคือ

- TMOD เป็นรีจิสเตอร์ขนาด 8 บิตที่ไม่สามารถเข้าถึงในระดับบิตได้โดยตรง ในการใช้งานนั้นจะต้อง เข้าไปใน รีจิสเตอร์ TMOD เพื่อกำหนดให้ ไทม์เมอร์/เคาน์เตอร์โหมดใดๆทำงาน โครงสร้างของ TMOD เป็นดังนี้

ตารางที่2.1 โครงสร้างของ TMOD

Bit no.	7	6	5	4	3	2	1	0
	Gate	C/T	M1	M0	Gate	C/T	M1	M0
	เป็นของ ไทม์เมอร์ / เคาน์เตอร์ 1				เป็นของ ไทม์เมอร์ / เคาน์เตอร์ 0			

จากโครงสร้างของ TMOD จะเห็นว่าเราสามารถใช้ TMOD กำหนดรูปแบบการทำงานของไทม์เมอร์/เคาน์เตอร์ 0 และ ไทม์เมอร์/เคาน์เตอร์ 1 ได้ในครั้งเดียวกัน โดยบิตสูง (บิตที่ 4-7) จะเอาไว้สำหรับกำหนดการทำงานของ ไทม์เมอร์/เคาน์เตอร์ 1 และบิตต่ำ (บิตที่ 0-3) จะทำหน้าที่ควบคุมการทำงานของ ไทม์เมอร์/เคาน์เตอร์ 0 แต่ละบิตมีความหมายดังนี้

ตารางที่2.2 ความหมายแต่ละบิตใน TMOD

ชื่อของบิต	หน้าที่
Gate	<u>ถ้าบิตนี้เป็น 0</u> หมายความว่าการทำงานของ ไทม์เมอร์/เคาน์เตอร์จะถูกควบคุมการทำงานโดยซอฟท์แวร์ คือจะสั่งการให้ ไทม์เมอร์/เคาน์เตอร์ทำงานหรือหยุดโดยการกำหนดค่าแก่บิต TRx ใน TCON <u>ถ้าบิตนี้เป็น 1</u> หมายความว่าเราต้องควบคุมการทำงานของ ไทม์เมอร์/เคาน์เตอร์ด้วยฮาร์ดแวร์ คือ ต้องกำหนดให้บิต TRx ใน TCON เป็น 1 และค่าสถานะของลอจิกที่ขา INT0 และ INT1 จะต้องเป็น 1 ตัว ไทม์เมอร์/เคาน์เตอร์จึงจะทำงาน

ตารางที่2.2 ความหมายแต่ละบิตใน TMOD (ต่อ)

ชื่อของบิต	หน้าที่
C/T	ถ้าบิตนี้เป็น 1 หมายความว่าให้ทำงานเป็นตัว เคาน์เตอร์
M1/M0	ถ้าบิตนี้เป็น 0 หมายความว่าให้ทำงานเป็นตัว ไทม์เมอร์
	เป็นบิตที่ใช้กำหนดว่าให้ ไทม์เมอร์/เคาน์เตอร์ทำงานเป็นโหมดใด ดังข้อกำหนดต่อไปนี้
	ถ้า M1=0, M0=0 แสดงว่าเลือกโหมด 0
	ถ้า M1=0, M0=1 แสดงว่าเลือกโหมด 1
	ถ้า M1=1, M0=0 แสดงว่าเลือกโหมด 2
	ถ้า M1=1, M0=1 แสดงว่าเลือกโหมด 3

- TCON เป็นรีจิสเตอร์ที่ทำหน้าที่ควบคุมการทำงานของ ไทม์เมอร์/เคาน์เตอร์ซึ่งรีจิสเตอร์ TCON นี้เป็นรีจิสเตอร์ขนาด 8 บิต ที่เราสามารถเข้าถึงการทำงานได้ในระดับบิต โดยรายละเอียดต่างๆ ในรีจิสเตอร์ TCON เป็นดังนี้

ตารางที่2.3 โครงสร้างของ TCON

Bit no.	7	6	5	4	3	2	1	0
	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

แต่ละบิตมีความหมายดังนี้

ตารางที่2.4 ความหมายแต่ละบิตใน TCON

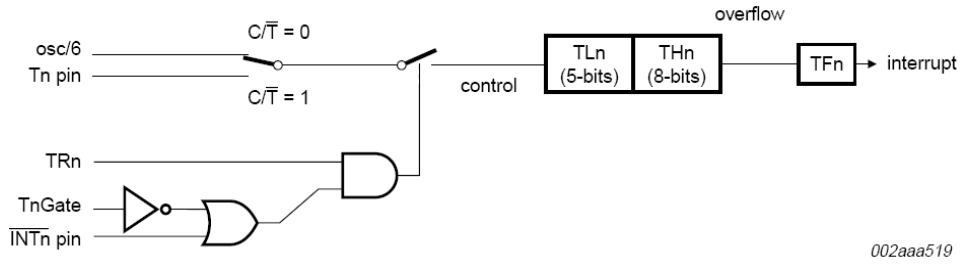
ชื่อของบิต	หน้าที่
IT0	เลือกลักษณะของสัญญาณอินเทอร์รัพท์ภายนอกตัวที่ 0 (INT0)
	ถ้าเป็น 0 เลือกให้ช่วงขอบขาลงเป็นสัญญาณอินเทอร์รัพท์
	ถ้าเป็น 1 เลือกให้ช่วงที่เป็นลอจิกต่ำเป็นสัญญาณอินเทอร์รัพท์
IE0	บิตนี้จะเป็น 1 เมื่อเกิดสัญญาณลักษณะเหมือนกับที่กำหนดใน IT0
IT1	เหมือนกับ IT0 แต่เป็นการกำหนดสำหรับ INT1
IE1	เหมือนกับ IE0 แต่เป็นการกำหนดสำหรับ INT1
TR0	เป็นบิตเปิดปิดการทำงานของ ไทม์เมอร์/เคาน์เตอร์ 0 คือ
	ถ้าเป็น 1 หมายถึงให้ ไทม์เมอร์/เคาน์เตอร์ 0 ทำงาน
	ถ้าเป็น 0 หมายความว่าหยุดการทำงานของ ไทม์เมอร์/เคาน์เตอร์ 0
TF0	ถ้าบิตนี้เป็น 1 แสดงว่าเกิดการ Overflow ของ ไทม์เมอร์/เคาน์เตอร์ 0
TR1	เหมือนกับ TR0 แต่ใช้กับ ไทม์เมอร์/เคาน์เตอร์ 1
TF1	เหมือนกับ TF0 แต่ใช้กับ ไทม์เมอร์/เคาน์เตอร์ 1

- TLx และ THx เป็นรีจิสเตอร์ขนาด 8 บิต ที่ทำหน้าที่เก็บข้อมูลจากการทำงานของไทม์เมอร์/เคาน์เตอร์ โดยที่ TL0/TH0 จะเป็นการเก็บข้อมูลของ ไทม์เมอร์/เคาน์เตอร์ 0 และ TL1/TH1 จะเป็นการเก็บข้อมูลของ ไทม์เมอร์/เคาน์เตอร์ 1

2.1.4.2 โหมดการทำงานของ ไทม์เมอร์/เคาน์เตอร์

ไทม์เมอร์/เคาน์เตอร์ มีโหมดการทำงานด้วยกัน 4 แบบคือ

- การทำงานในโหมด 0 เป็นการนับแบบ 13 บิต โดยใช้ TLn (TL0 หรือ TL1) จำนวน 5 บิตล่างกับ THn (TH0 กับ TH1) จำนวน 8 บิต ในการนับ ในการนับนั้นเมื่อ TLn มีค่าเป็น 00011111_2 แล้วมีการนับเพิ่มอีก 1 ตัว TLn ก็จะส่งบิต 1 ไปให้ THn นับต่อ (THn มีค่าเพิ่มขึ้นอีก 1) แล้ว TLn จะเปลี่ยนค่าเป็น 00000000_2 แล้วทำการนับต่อไปเรื่อยๆ จนกว่า TLn เป็น 00011111_2 และ THn เป็น 11111111_2 ก็จะเกิดการ Overflow กับ TFn (TF0 หรือ TF1) หลังจากนั้นไมโครคอนโทรลเลอร์ก็จะสร้างสัญญาณอินเทอร์รัพท์ สำหรับไทม์เมอร์/เคาน์เตอร์ ขึ้นมา แล้วเปลี่ยนค่าของ TLn กับ THn ให้เป็น 00000000_2 เพื่อเริ่มต้นการนับต่อไป ดังในภาพที่ 2.5



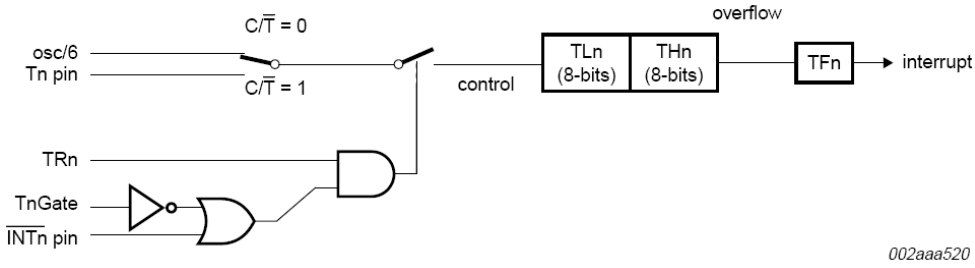
ภาพที่2.5 ไทม์เมอร์/เคาน์เตอร์ 0 หรือ 1 ในโหมด 0 (13-bit เคาน์เตอร์)

- การทำงานในโหมด 1 เป็นการนับแบบ 16 บิต โดยใช้ TLn เก็บข้อมูลไบต์ล่าง และ THn เก็บข้อมูลไบต์สูง ดังนี้

ตารางที่2.5 การทำงานในโหมด 1

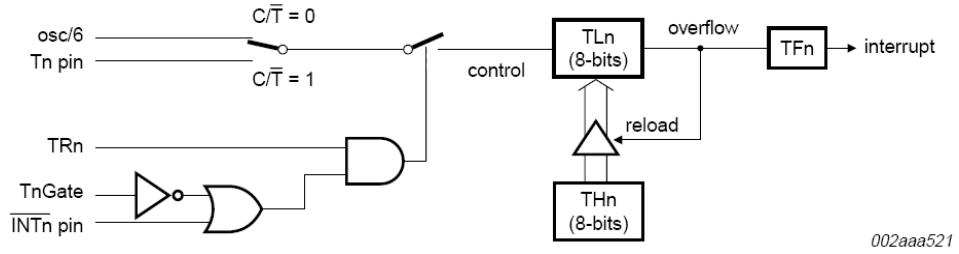
Bit no.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	THn								TLn							

เวลาทำงานก็จะเริ่มนับค่าจาก 0000_{16} หรือจากค่าที่ตั้งเอาไว้ไปจนกว่าจะถึง $FFFF_{16}$ หลังจากนั้นก็จะเกิดการ Overflow กับ TFn หลังจากนั้นไมโครคอนโทรลเลอร์ก็จะทำการสร้างสัญญาณ อินเทอร์รัพท์สำหรับไทม์เมอร์/เคาน์เตอร์ ขึ้นมา แล้วเปลี่ยนค่าของ TLn กับ THn ให้เป็น 0000_{16} เพื่อเริ่มต้นการนับต่อไป ดังภาพที่ 2.6



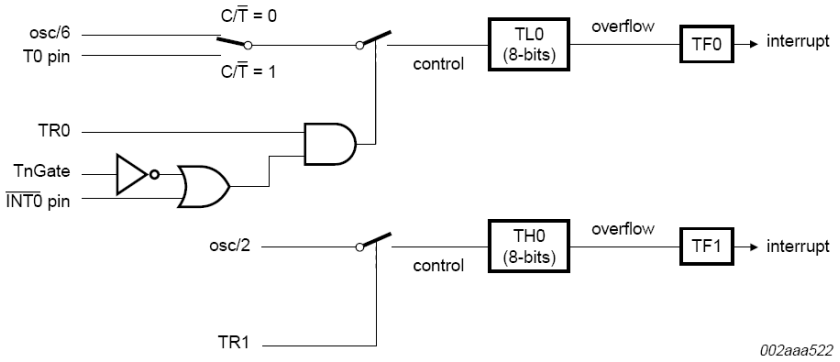
ภาพที่2.6 ไทม์เมอร์/เคาน์เตอร์ 0 หรือ 1 ในโหมด 1 (16-bit counter)

- การทำงานในโหมด 2 เป็นการนับแบบ 8 บิต โดยใช้ TLn เป็นตัวนับ และ THn เป็นตัวเก็บค่าเริ่มต้น เมื่อ TLn นับจนมีค่าเป็น FF_{16} ก็จะเกิดการ Overflow กับ TFn หลังจากนั้นไมโครคอนโทรลเลอร์จะทำการสร้างสัญญาณอินเทอร์รัพท์สำหรับ ไทม์เมอร์/เคาน์เตอร์ ขึ้นมา แล้วเปลี่ยนค่าของ TLn ให้เป็นค่าที่เก็บเอาไว้ใน THn เพื่อเริ่มต้นการนับต่อไป ดังภาพที่ 2.7



ภาพที่2.7 ไทม์เมอร์/เคาน์เตอร์ 0 หรือ 1 ในโหมด 2 (8-bit auto-reload)

- การทำงานในโหมด 3 เป็นการนับแบบ 8 บิต ที่แยกการนับของ TH0 กับ TL0 ออกจากกัน แล้วใช้ TF1 เป็นที่เก็บการเกิด Overflow ของ TH0 และใช้ TF0 เป็นที่เก็บการเกิด Overflow ของ TL0 การทำงานในโหมดนี้ จะทำให้ ไทม์เมอร์/เคาน์เตอร์ 1 ไม่สามารถสร้างสัญญาณอินเทอร์รัพท์ได้ (เพราะ ไทม์เมอร์/เคาน์เตอร์ 0 ได้ใช้ TF1 ไปแล้ว) ดังภาพที่ 2.8



ภาพที่2.8 ไทม์เมอร์/เคาน์เตอร์0 หรือ 1 ในโหมด 3 (8-bit เคาน์เตอร์ จำนวน2ตัว)

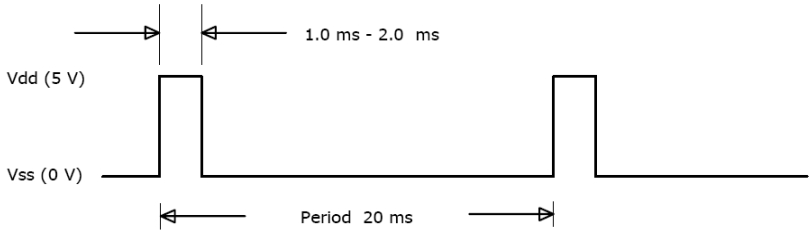
2.2 อาร์.ซี.เซอร์โวมอเตอร์ (RC Servo motor)

อาร์.ซี.เซอร์โวมอเตอร์ คือ มอเตอร์ไฟฟ้ากระแสตรง (DC motor) ที่ถูกประกอบรวมกับ ชุดเกียร์ และ ส่วนควบคุม ต่างๆ ไว้ในโมดูลเดียวกัน หรือ ภายในกล่องพาสติกเดียวกัน โดยมอเตอร์ชนิดนี้จะมีสายต่อใช้งานเพียง 3 เส้นเท่านั้น คือ VCC, GND และ สายสัญญาณควบคุม(Control Line) ซึ่งสามารถควบคุมให้มอเตอร์หมุนซ้าย หรือขวาได้จากสายสัญญาณเพียงเส้นเดียว โดยสัญญาณที่ใช้ควบคุมนี้ จะเป็นสัญญาณพัลส์วิดมอดูเลต (Pulse Width Modulation, PWM) แบบ TTL Level ระดับแรงดันที่จ่ายให้มอเตอร์นี้จะอยู่ในช่วงประมาณ 4 ถึง 6 โวลท์ ขึ้นอยู่กับคุณสมบัติของมอเตอร์แต่ละตัว ข้อดีของมอเตอร์ชนิดนี้ ก็คือ จะมีขนาดเล็กน้ำหนักเบา, ให้แรงบิตสูง, กินพลังงานน้อย และ สามารถควบคุม ด้วยแรงดันลอจิกที่เป็น TTL ได้โดยตรงไม่จำเป็นต้องต่อวงจรขับ (Driver) อื่นๆ เพราะ มอเตอร์ชนิดนี้จะมีวงจรควบคุมบรรจุไว้ภายในอยู่แล้ว ซึ่งมอเตอร์ชนิดนี้สามารถควบคุมให้หมุนไปในตำแหน่ง หรือ ทิศทางองศาที่ต้องการได้ โดยอาศัยสัญญาณความกว้างพัลส์ที่ป้อนให้มอเตอร์ แต่เซอร์โว-มอเตอร์นี้จะหมุนได้แค่เพียงในช่วงประมาณ 180° หรือ ครึ่งรอบเท่านั้น หรือ บางรุ่นอาจหมุนได้ถึง 210° แต่จะไม่สามารถหมุนเป็นวงรอบได้ เนื่องจากโครงสร้างภายในจะประกอบด้วย ตัวต้านทานชนิดปรับค่าได้ (Variable Resister) ที่ทำหน้าที่ตรวจสอบตำแหน่งการหมุนของมอเตอร์ และ ตัวต้านทานนี้จะถูกยึดติดกับแกนหมุนของมอเตอร์ ซึ่งจากการที่ตัวต้านทานปรับค่านี้ ไม่สามารถหมุนเป็นวงรอบได้ ดังนั้น เซอร์โวมอเตอร์จึงถูกออกแบบให้หมุนได้เพียงแค่ประมาณ 180 องศา หรือ ครึ่งรอบเท่านั้น เพื่อป้องกันความเสียหายที่จะเกิดกับตัวต้านทานปรับค่าได้

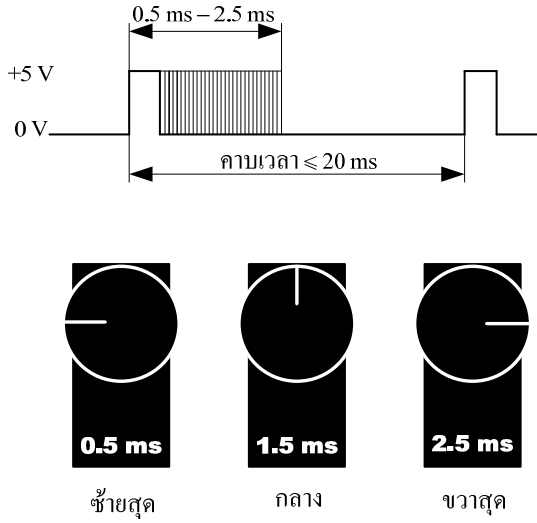
ภาพที่2.9 ส่วนประกอบต่างๆของ อาร์.ซี.เซอร์โวมอเตอร์

2.2.1 หลักการทำงาน อาร์.ซี.เซอร์โวมอเตอร์

การควบคุมการทำงาน ของเซอร์โวมอเตอร์ ทำได้โดย การป้อนสัญญาณความกว้างของพัลส์ ให้กับมอเตอร์ซึ่งตำแหน่งและทิศทางการหมุนของมอเตอร์นี้จะขึ้นอยู่กับขนาดของความกว้างของพัลส์นั้นๆ ดังภาพที่ 13. โดยทั่วไปแล้วความกว้างของสัญญาณพัลส์จะมีจุดให้อ้างอิง 3 จุด ดังภาพที่ 2.11



ภาพที่2.10 ลักษณะสัญญาณความกว้างของพัลส์



ภาพที่2.11 ตำแหน่งการหมุนมอเตอร์

- สัญญาณความกว้างพัลส์ขนาด 1.5 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม 0 องศา หรือ จุดกึ่งกลางของมอเตอร์
- สัญญาณความกว้างพัลส์ขนาด 1 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม -90 องศา หรือ ในทิศทางทวนเข็มนาฬิกา
- สัญญาณความกว้างพัลส์ขนาด 2 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม +90 องศา หรือในทิศทางตามเข็มนาฬิกา

ค่าความกว้างพัลส์ และ ระยะองศาการหมุนของมอเตอร์ที่อธิบายด้านบนนั้น เป็นเพียงค่าประมาณเท่านั้น ทั้งนี้ระยะการหมุน และ ขนาดของพัลส์ที่ควบคุมการทำงานของมอเตอร์ ในแต่ละยี่ห้ออาจจะไม่เท่ากัน

ดังนั้น ในการใช้งานจึงควรศึกษารายละเอียดของมอเตอร์ในแต่ละรุ่นที่นำมาใช้ ซึ่งโดยปกติแล้วรายละเอียดต่างๆ ของมอเตอร์ มักจะมีติดมากับตัวมอเตอร์นั้นๆ อยู่แล้ว

ในการที่จะควบคุมให้มอเตอร์หมุนเป็นมุมอื่นๆ นั้นก็สามารถทำได้โดยการป้อนสัญญาณพัลส์เป็นระดับความกว้างต่างๆ โดยอ้างอิงกับจากจุด ทั้ง 3 จุดที่กล่าวมานี้ ตัวอย่างเช่น ถ้าต้องการให้มอเตอร์หมุนไปที่มุม – 45 องศา เราก็จะต้องป้อนสัญญาณพัลส์ที่มีความกว้าง 1.25 ms เป็นต้น และสัญญาณพัลส์นี้จะต้องจ่ายให้มอเตอร์ทุกๆ 20 ms (Period) เพื่อรักษาสภาพตำแหน่งของมอเตอร์ไว้

โดยหลักการก็คือ จะอาศัยการเปรียบเทียบช่วงเวลาของความกว้างของพัลส์ที่จ่ายให้กับมอเตอร์ทางขาสัญญาณควบคุมกับค่าเวลาของวงจร RC ภายในบอร์ดควบคุมในตัวของมอเตอร์ ซึ่งค่าเวลาของวงจร RC นี้จะมีการเปลี่ยนแปลงตามการหมุนของมอเตอร์ เนื่องจากตัวต้านทานปรับค่าจะถูกยึดติดอยู่กับแกนหมุนของมอเตอร์ ซึ่งการหมุนของมอเตอร์จะทำให้ค่าความต้านทานของตัวต้านทานปรับค่า (VR) เปลี่ยนแปลงไป เป็นผลทำให้ค่าเวลาของวงจร RC เปลี่ยนแปลงตามไปด้วย โดยในขณะที่เราป้อนสัญญาณความกว้างของพัลส์ให้กับมอเตอร์ทางขาสัญญาณควบคุม สัญญาณนี้จะถูกนำไปเปรียบเทียบกับค่าเวลาของวงจร RC หากค่าทั้ง 2 ไม่เท่ากันมอเตอร์ก็จะหมุนทำให้ค่าเวลาของวงจร RC เปลี่ยนแปลงจนกระทั่งค่าเวลาความกว้างพัลส์ของ วงจร RC เปลี่ยนแปลงจนเท่ากับสัญญาณพัลส์ทางขาควบคุม (Control Line) มอเตอร์จึงจะหยุดหมุน

2.3 จลนศาสตร์ (Kinematics)

จลนศาสตร์คือ การศึกษาทางด้านเรขาคณิต (geometry) โดยเฉพาะการเคลื่อนที่ของแขนกลในหุ่นยนต์ โดยไม่คิดถึงแรง (force) รูปร่าง (shape) ขนาด (size) และน้ำหนัก (mass properties)

ปัญหาทางจลนศาสตร์ของหุ่นยนต์มี 2 แบบคือ จลนศาสตร์แบบไปข้างหน้า (Forward kinematics) และจลนศาสตร์แบบผกผัน (Inverse Kinematics) ซึ่งจะนำมาใช้ในการกำหนดมุมในแต่ข้อต่อของหุ่นยนต์ ในแต่ละจังหวะการเคลื่อนที่

2.3.1 จลนศาสตร์แบบไปข้างหน้า (Forward Kinematics)

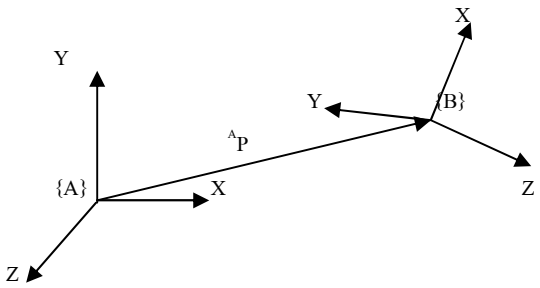
ในการคำนวณการหาสมการพื้นฐานของโครงสร้างของแขนหุ่นนั้น เราจะต้องกำหนดแกนสมมุติขึ้นมา (X-Y-Z) ซึ่งเราจะได้ตัวแปรที่ไม่ทราบค่าของแต่ละข้อต่อ เพื่อหาตำแหน่งปลายสุดของแขนหุ่นยนต์ จะบ่งบอกถึง ตำแหน่ง (${}^{A}P$, Position vector)

$$ {}^{A}P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \tag{2.1} $$

และลักษณะการหมุนของแกนข้อต่อ เมื่อเปรียบเทียบกับแกนข้อต่อที่ผ่านมา (${}^{A}_{B}R$, Rotation matrix)

$$ {}^{A}_{B}R = \begin{bmatrix} {}^{A}\bar{X}_B & {}^{A}\bar{Y}_B & {}^{A}\bar{Z}_B \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{2.2} $$

$$r_{11} = X_B \cdot X_A \quad ; \quad r_{12} = Y_B \cdot X_A \quad ; \quad r_{13} = Z_B \cdot X_A$$
$$r_{21} = X_B \cdot Y_A \quad ; \quad r_{22} = Y_B \cdot Y_A \quad ; \quad r_{23} = Z_B \cdot Y_A$$
$$r_{31} = X_B \cdot Z_A \quad ; \quad r_{32} = Y_B \cdot Z_A \quad ; \quad r_{33} = Z_B \cdot Z_A$$



ภาพที่2.12 ความสัมพันธ์ระหว่างเวคเตอร์ กับแกน

เมื่อมีการหมุน หรือเลื่อนตำแหน่งของข้อต่อจะมีการเปลี่ยนรูปของแกนใหม่ ดังนั้นเราสามารถเขียนให้อยู่ในรูปของการเปลี่ยนรูปแบบ (Transformation operators) ได้โดยการ กำหนด ${}^{A}_{B}R$ (Rotation matrix) ให้อยู่ในรูปแบบ 4×4 และ ${}^{A}P$ (Position vector) ให้อยู่ในรูปแบบ 4×1

$${}^{A}P = {}^{A}_{B}T \; {}^{B}P \tag{2.3}$$

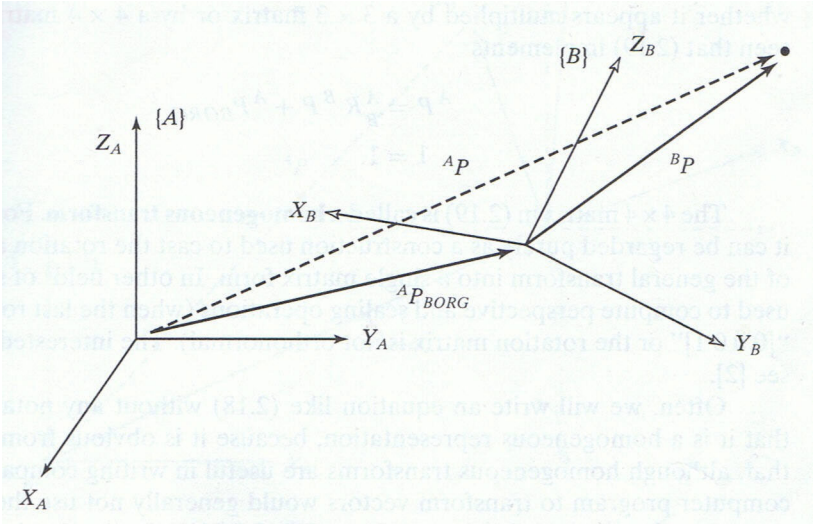
จากสามการข้างต้นเราสามารถทำให้อยู่ในรูป Homogeneous Transform

$$\begin{bmatrix} {}^{A}P \\ 1 \end{bmatrix} = \left[\begin{array}{c|c} {}^{A}_{B}R & {}^{A}P_{BORG} \\ \hline 0\ 0\ 0 & 1 \end{array}\right] \begin{bmatrix} {}^{B}P \\ 1 \end{bmatrix} \tag{2.4}$$

การเปลี่ยนรูปแบบทั่วไป (General Transformation)

$${}^{i-1}_{i}T = \left[\begin{array}{ccc|c} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1} d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1} d_i \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \tag{2.5}$$

ในที่นี้เราจะกำหนดให้ $\sin\theta_i = s_i$, $\cos\theta_i = c_i$



ภาพที่2.13 รูปแบบทั่วไปของการแปลงเวคเตอร์

ในการกำหนดทิศทางของแกนนั้น เราจะกำหนดให้แกนนั้นอยู่ตามจุดข้อต่อต่างๆ เช่นข้อต่อแบบหมุนได้ จะกำหนดให้แกน Z เป็นแกนหมุน ดังในภาพที่ 2.14

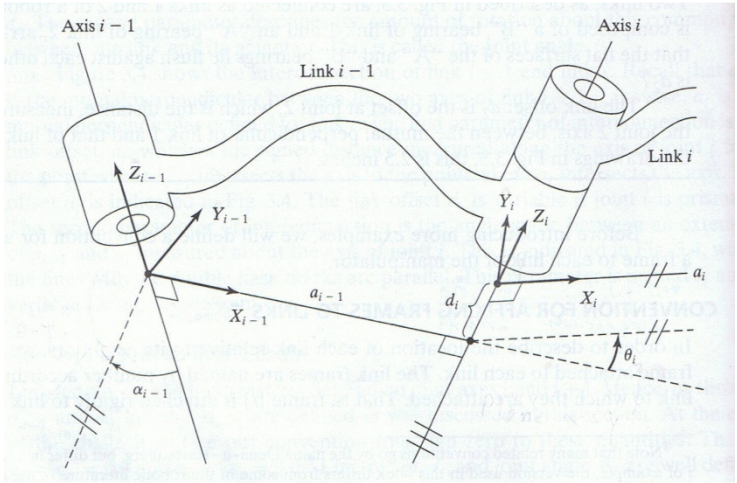
ค่าตัวแปรที่เกิดขึ้นระหว่างแกน i-1 กับ i

a_i = ระยะทางจาก Z_i ถึง Z_{i+1} รอบแกน X_i

α_i = องศาของมุมจาก Z_i ถึง Z_{i+1} รอบแกน X_i

d_i = ระยะทางจาก X_i ถึง X_{i+1} รอบแกน Z_i

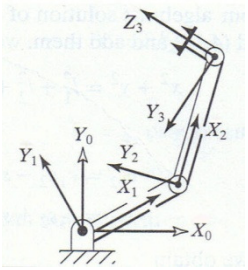
θ_i = องศาของมุมจาก X_i ถึง X_{i+1} รอบแกน Z_i



ภาพที่2.14 แกนข้อต่อ

<u>ตัวอย่าง</u> การหาสมการหุ่นยนต์ระดับความเป็นอิสระ 3 (3 DOF)

ในการคำนวณหาตำแหน่งของจุดปลายแขนของหุ่นยนต์นั้น เมื่อเราทราบค่าความยาวของแขนหุ่นยนต์แต่ละช่วง เราสามารถหาสมการพื้นฐานได้ดังนี้



ภาพที่2.15 3 องศาอิสระ

เมื่อเรากำหนดแกนในแต่ละข้อต่อแล้ว เราจะได้ค่าตัวแปรต่างๆดังนี้

ตารางที่2.6 ตาราง Denavit-Hartenberg

i	$\alpha_i - 1$	$a_i - 1$	d_i	θ_i
1	0	0	0	θ_1
2	0	L_1	0	θ_2
3	0	L_2	0	θ_3

จากนั้นหาการเปลี่ยนรูป Transformations Matrix)ในแต่ละข้อต่อ

$$ {}^{0}_{1}\mathrm{T} = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{1}_{2}\mathrm{T} = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & L_1 \\ s\theta_2 & c\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{2}_{3}T = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & L_2 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

ความสัมพันธ์ระหว่างข้อต่อที่1 กับข้อต่อที่2 (${}^{0}_{2}T$)

ดังนั้น ${}^{0}_{2}T = {}^{0}_{1}T \bullet {}^{1}_{2}T$

$$ {}^{0}_{2}T = \begin{bmatrix} c_{12} & -s_{12} & 0 & L_1c_1 \\ s_{12} & c_{12} & 0 & L_1s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

ดังนั้น ${}^{0}_{3}T = {}^{0}_{2}T \bullet {}^{2}_{3}T$

$$ {}^{0}_{3}T = \begin{bmatrix} c_{123} & -s_{123} & 0 & L_1c_1+L_2c_{12} \\ s_{123} & c_{123} & 0 & L_1s_1+L_2s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

สุดท้ายเราจะได้ สมการการหมุนของแขนหุ่นยนต์ และตำแหน่งปลายสุดของหุ่นยนต์

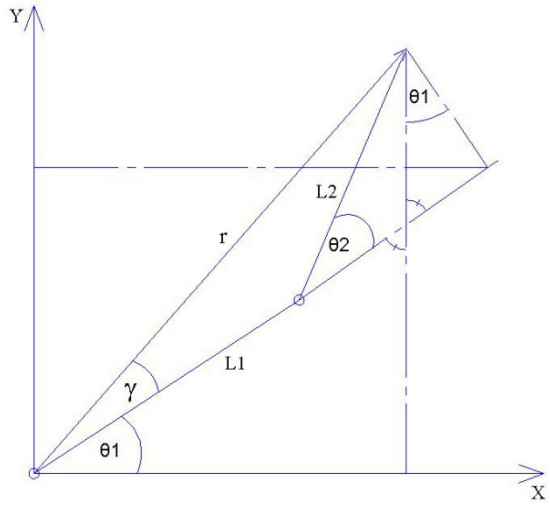
$$ Px = L_1c_1+L_2c_{12} \qquad Py = L_1s_1+L_2s_{12} \qquad Pz = 0 \tag{2.6} $$

2.3.2 จลนศาสตร์แบบผกผัน (Inverse kinematics)

จลนศาสตร์แบบผกผัน คือ การคำนวณหาค่าตัวแปรของแต่ละข้อต่อโดยการกำหนดตำแหน่งที่ปลายของแขนกล วิธีการคำนวณจลนศาสตร์แบบผกผันนี้ ในบางครั้งมีได้หลายคำตอบ หรือไม่สามารถหาคำตอบได้ การคำนวณนี้ค่อนข้างยุ่งยากกว่าการคำนวณจลนศาสตร์แบบไปข้างหน้า ซึ่งคำตอบของวิธีการคำนวณจลนศาสตร์ผกผันนี้อาจแบ่งได้ 2 รูปแบบคือ closed form และ numerical iterative form ซึ่งโดยรูปแบบของ closed form นั้นเราสามารถหาคำตอบให้ออกมาในรูปแบบของฟังก์ชั่น ซึ่งจะง่ายต่อการคำนวณหาค่าเพราะสามารถแทนค่าในตัวแปรของฟังก์ชั่นนั้น

ส่วนวิธีแบบการทำซ้ำเชิงตัวเลข (Numerical iterative form) นั้นใช้วิธีสมมติค่าตัวเลข เริ่มต้นแล้วทำซ้ำไปเรื่อยๆ จนกว่าค่านั้นไม่มีการเปลี่ยนแปลง หรือเปลี่ยนแปลงน้อยมาก หรืออีกนัยหนึ่งคือ ค่านั้นลู่สู่ค่าหนึ่งซึ่งเป็นการยากต่อการคำนวณเพราะในบางครั้งอาจหาคำตอบไม่ได้

ตัวอย่าง การหาสมการจลนศาสตร์แบบผกผัน แขนกล 3 องศาอิสระ (3 DOF)



ภาพที่2.16 แสดงตำแหน่ง ของแขนกล 3 องศาอิสระ

จากการหาจลนศาสตร์แบบไปข้างหน้า เราจะทราบค่าของตำแหน่งของปลายแขนดังนี้

$$c_\phi = c_{123}$$

$$s_\phi = s_{123}$$

$$Px = L_1c_1+L_2c_{12}$$

$$Py = L_1s_1+L_2s_{12} \tag{2.7}$$

ยกกำลังสอง Px และ Py จะได้

$$Px^2 + Py^2 = L_1^2 + L_2^2 + 2L_1L_2c_2 \tag{2.8}$$

เมื่อให้

$$c_{12} = c_1c_2 - s_1s_2$$

$$s_{12} = c_1s_2 + s_1c_2$$

จะได้ค่าของ c_2

$$c_2 = \frac{Px^2+Py^2-L_1^2-L_2^2}{2L_1L_2}$$

$$s_2 = \sqrt{1-c_2^2}$$

$$\theta_2 = \text{Atan2}(s_2, c_2) \tag{2.9}$$

จากภาพที่2.16 จะได้

$$Px = k_1c_1 - k_2s_2$$

$$Py = k_1s_1 + k_2c_1 \tag{2.10}$$

$$k_1 = L_1 + L_2c_2$$

$$k_2 = L_2 s_2 \tag{2.11}$$

$$r = \sqrt{k_1^2 + k_2^2} \tag{2.12}$$

$$\gamma = \text{Atan2}(k_2 , k_1) \tag{2.13}$$

$$k_1 = r \cos \gamma$$

$$k_2 = r \sin \gamma \tag{2.14}$$

$$\frac{Px}{r} = \cos \gamma \cos \theta_1 - \sin \gamma \sin \theta_1$$

$$\frac{Py}{r} = \cos \gamma \sin \theta_1 + \sin \gamma \cos \theta_1 \tag{2.15}$$

ดังนั้น

$$\cos (\gamma + \theta_1) = \frac{Px}{r}$$

$$\sin (\gamma + \theta_1) = \frac{Py}{r}$$

$$\gamma + \theta_1 = \text{Atan2} (\frac{Py}{r}, \frac{Px}{r}) = \text{Atan2} (Py , Px)$$

$$\theta_1 = \text{Atan2} (Py , Px) - \text{Atan2} (k_2 , k_1) \tag{2.16}$$

จากที่ $c_\phi = c_{123}$ เราจะได้

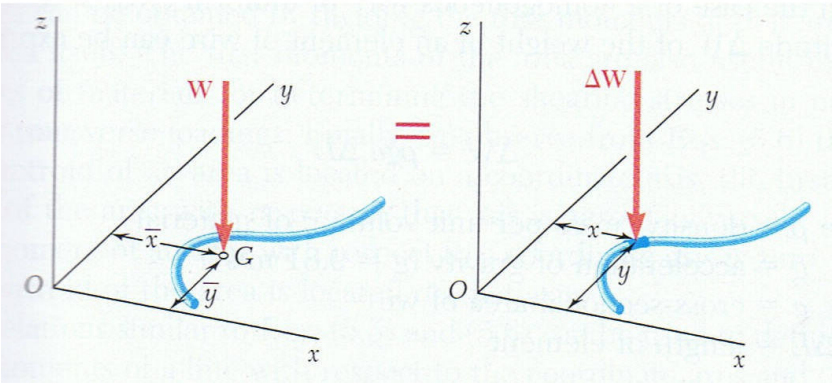
$$\theta_1 + \theta_2 + \theta_3 = \text{Atan2} (s_\phi , c_\phi) = \phi \tag{2.17}$$

โดยที่ Atan2θ คือค่าความลาดเอียงของเส้นที่ลากมาจากพิกัดมายังจุดกำเนิด

2.4 จุดศูนย์ถ่วงของเส้น (Centroid of line)

เมื่อเราวางวัตถุที่เป็นเส้น ที่มีขนาดเท่ากันตลอดทั้งเส้นในแนวราบ ไม่จำเป็นว่าจุดศูนย์ถ่วงของวัตถุนั้นจะต้องอยู่ภายในของวัตถุนั้น

เมื่อคิดน้ำหนัก (W = mg) ที่จุดศูนย์ถ่วงมวล ที่ทำกับแนวแกนจะก่อให้เกิดโมเม้นรอบ แกน x และ y ขึ้นมา ในขณะเดียวกันเมื่อแบ่งคิดน้ำหนักในแต่ละส่วนของเส้น (ΔW) ที่ทำกับแนวแกนก็สามารถทำให้เกิด โมเม้นย่อยรอบ แกน x และ y ได้เช่นเดียวกัน



ภาพที่2.17 จุดศูนย์กลางแรงดึงดูดของเส้น

จากภาพที่2.17 เราจะได้สมการดังนี้

$$\sum M_y \quad : \quad \bar{x}W = \sum x\,\Delta W$$
$$\sum M_x \quad : \quad \bar{y}W = \sum y\,\Delta W \qquad (2.18)$$

ในกรณีที่ลักษณะของเส้นนั้นมีพื้นที่หน้าตัดเท่ากันตลอดทั้งเส้น และเป็นเนื้อวัสดุชนิดเดียวกัน น้ำหนักของเส้นหาได้จาก

$$\Delta W = \rho \cdot g \cdot a \cdot \Delta L \qquad (2.19)$$

ρ = ความหนาแน่น (kg/m^3)

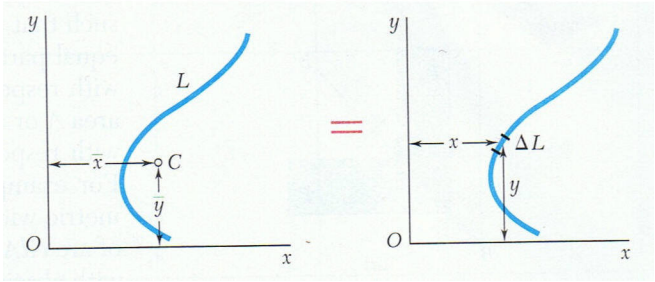
g = แรงดึงดูดของโลก (m/s^2)

a = พื้นที่หน้าตัดของเส้น (m^2)

ΔL = ความยาวย่อยของเส้น (m)

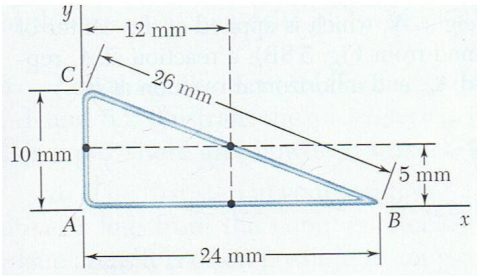
เพราะฉนั้นเมื่อค่าความหนาแน่น แรงดึงดูด และพื้นที่หน้าตัด มีค่าคงที่เท่ากันตลอดทั้งเส้น เราสามารถหาค่าจุดศูนย์ถ่วงได้ (ภาพที่2.18) ดังนี้

$$\sum M_y \quad : \quad \bar{x}L = \sum x\,\Delta L$$
$$\sum M_x \quad : \quad \bar{y}L = \sum y\,\Delta L \qquad (2.20)$$



ภาพที่2.18 จุดศูนย์ถ่วง

<u>ตัวอย่าง</u> การหาจุดศูนย์ถ่วงของเส้นที่ยาวต่อเนื่องกัน



ภาพที่2.19 ตัวอย่างการหาจุดศูนย์ถ่วงของเส้น

ตารางที่2.7 ตัวอย่างการหาจุดศูนย์ถ่วง

ส่วนที่	L.	$\bar{x}$	$\bar{y}$	$\bar{x}L$	$\bar{y}L$
AB	24	12	0	288	0
BC	26	12	5	312	130
CA	10	0	5	0	50
	$\sum L = 60$			$\sum \bar{x} L = 600$	$\sum \bar{y} L = 180$

นำค่าที่ได้จากตารางมาเข้าสู่สมการหาค่าจุดศูนย์ถ่วง

$\bar{X} \sum L = \sum \bar{x} L$; $\bar{X} (60) = 600$ จุดศูนย์ถ่วง $\bar{X} = 10$ mm

$\bar{Y} \sum L = \sum \bar{y} L$; $\bar{Y} (60) = 180$ จุดศูนย์ถ่วง $\bar{Y} = 3$ mm