บทที่ 3

การพัฒนาระบบแขนกลควบคุมระยะไกลสำหรับการจัดเปลี่ยนต้นกำเนิดรังสีความแรงสูง

3.1 ข้อมูลพื้นฐานของการออกแบบแขนกล

ข้อมูลสำหรับออกแบบในขั้นต้นนี้จะคำนึงถึงลักษณะในการทำงาน ความสามารถในการจับวัตถุ การเคลื่อนที่ในทิศทางต่างๆ ขอบเขตการทำงานของมือระบบแขนกลที่พัฒนาขึ้นจึงมีส่วนที่จะต้องพิจารณา 4 ส่วน คือ

ก. ส่วนประกอบทางกล การเคลื่อนที่ของแขนกลเลือกการเคลื่อนที่ในลักษณะ Articulated Configuration เนื่องจากควบคุมการเคลื่อนของข้อต่อได้ละเอียดโดยมีพิกัดของกลไกดังนี้

ก.1 แขนหมุนรอบฐาน รัศมี 360 องศา

ก.2 มุมในการหมุนข้อต่อไหล่ (60 องศา)

ก.3 มุมในการหมุนข้อต่อศอก (60 องศา)

ก.4 มุมในการหักข้อมือขึ้น-ลง (60 องศา)

ก.5 มุมในการหมุนข้อมือโดยรอบ (15 องศา)

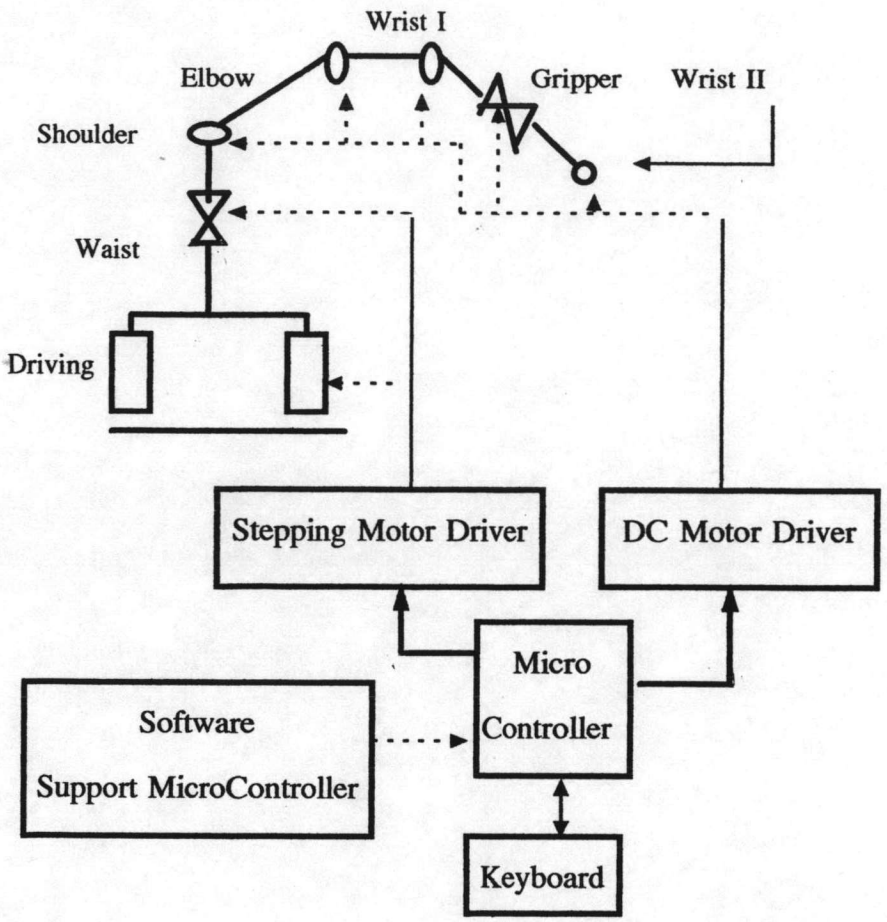
ก.6 มือจับกางปากได้กว้าง 6 ซม.

ข. ส่วนประกอบระบบขับเคลื่อน เลือกแบบระบบไฟฟ้าซึ่งใช้มอเตอร์เป็นต้นกำลังในการขับเคลื่อนส่งกำลังไปยังอุปกรณ์กลต่างๆด้วยวงจรสวิตช์และรีเลย์ ส่วนที่ต้องการแรงบิดสูงใช้มอเตอร์กระแสตรง ส่วนที่ต้องการเคลื่อนที่ละเอียดใช้มอเตอร์สเต็ปปิง

ค. ส่วนประกอบทางอิเล็กทรอนิกส์ ได้แก่ระบบขับรีเลย์และโซลินอยด์ ระบบเชื่อมโยงสัญญาณจากไมโครคอนโทรลเลอร์ เพื่อประสานการทำงานจากโปรแกรมควบคุม

ง. โปรแกรมควบคุมการทำงาน เลือกการควบคุมการเคลื่อนตำแหน่งและทิศทางของข้อต่อและอุปกรณ์กลผ่านไมโครคอนโทรลเลอร์ โดยออกแบบโปรแกรมสนับสนุนการใช้งาน ด้วยการกำหนดฟังก์ชันบนคีย์บอร์ด

ส่วนประกอบของแขนกลทั้ง 4 ส่วน มีการทำงานเชื่อมโยงกันดังในแผนภาพรูปที่ 3.1



รูปที่ 3.1 แผนภาพระบบแขนกลที่ออกแบบขึ้น

3.2 การออกแบบระบบทางกล

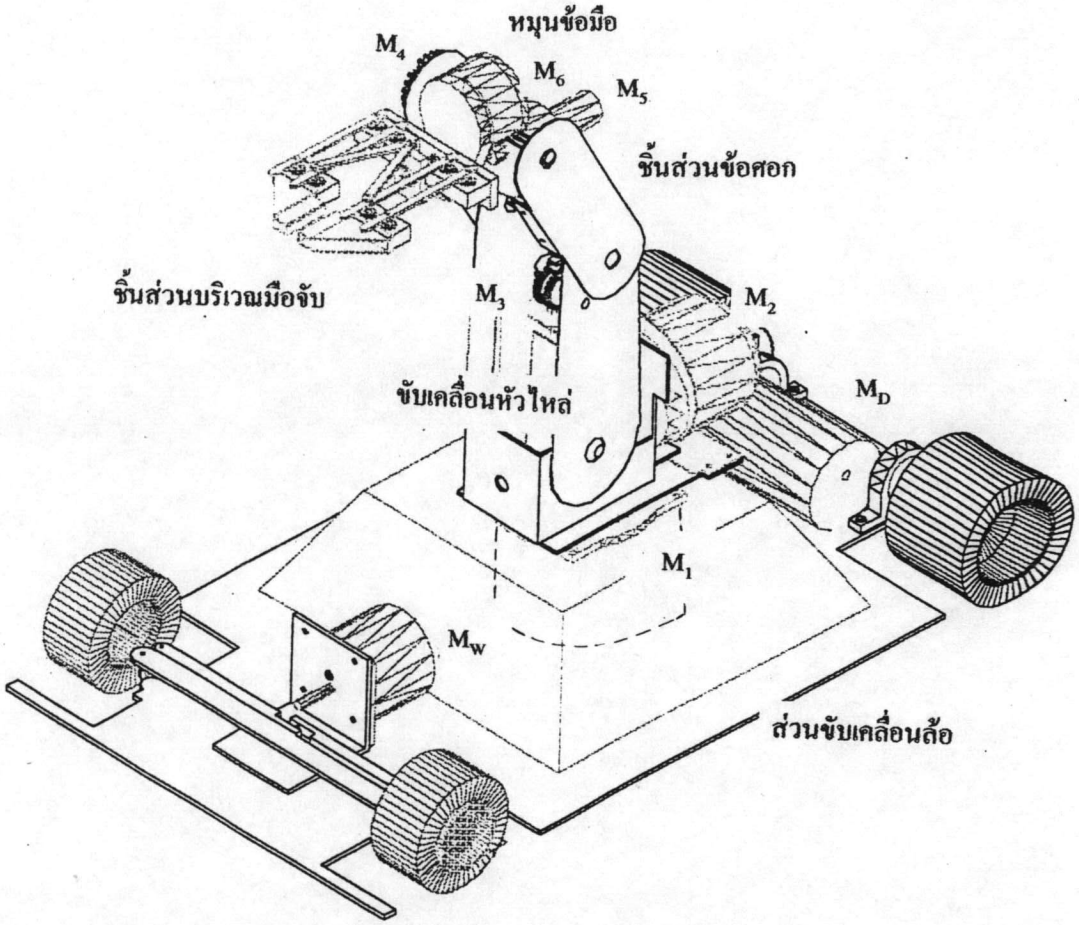
แขนกลที่ออกแบบนี้จะเป็นแบบ 5 ดีกรีออฟฟรีดอม (Degree of freedom) โดยจำนวนข้อต่อจะเคลื่อนที่เป็นอิสระต่อกัน การทำงานของแขนกลจะประกอบด้วยการเคลื่อนไหวของฐาน, ไหล่, ข้อศอก, ข้อมือพร้อมด้วยระบบขับเคลื่อนล้อ ดังโครงสร้างในรูปที่ 3.2 และแบบชิ้นส่วนรายละเอียดในภาคผนวก ก.

3.2.1 ส่วนขับเคลื่อนล้อ

ส่วนขับเคลื่อนล้อ ทำงานเป็นระบบขนส่งแขนกลไปสู่เป้าหมายที่จะทำงาน ประกอบด้วยตัวส่งกำลังขับเคลื่อน และตัวกำหนดทิศทางการขับเคลื่อนโดย M_w เป็น

I 17432170

มอเตอร์สเต็ปปิงส่งกำลังผ่านเพลาซึ่งบังคับทิศทาง 2 ล้อหน้า M_D เป็นมอเตอร์สเต็ปปิง ขับผ่านสายพานสู่ล้อขับเคลื่อน 2 ล้อหลัง



รูปที่ 3.2 โครงสร้างของแขนกลที่พัฒนาขึ้น

3.2.2 <u>ชิ้นส่วนกลบริเวณหัวไหล่</u>

M_1 เป็นมอเตอร์สเต็ปปิงยึดกับฐานแขนกล ส่งกำลังโดยตรงจากเพลาให้ชุดหัวไหล่ แขนกลส่วนนี้จะสามารถหมุนได้รอบตัว ชุดหัวไหล่มีมอเตอร์กระแสตรง M_2 ตั้งฉากกับ M1 เพลาของ M_2 จะถูกขับผ่านเกลียวหนอน (Worm Shaft) ที่ใช้ขับเฟืองหนอน (Worm Gear) โดยเพลาสองเพลาทำมุมฉากต่อกัน และทำหน้าที่ขับเคลื่อนชิ้นส่วนหัวไหล่

3.2.3 ชิ้นส่วนกลบริเวณข้อศอก

M_3 มอเตอร์กระแสตรงยึดกับส่วนกลหัวไหล่ ส่งกำลังจากเพลามอเตอร์ ขับผ่านเกลียวหนอนปากเดียว ส่งกำลังไปขับเฟืองหนอนในส่วนของชุดข้อศอก

3.2.4 ชิ้นส่วนกลบริเวณข้อมือยก ขึ้น-ลง

M_4 มอเตอร์กระแสตรงยึดอยู่ด้านนอกปีกของส่วนกลข้อศอก ส่งกำลังผ่านเพลาของมอเตอร์ สู่เกลียวหนอนปากเดียวขับให้กับเฟืองหนอน ทำให้เพลาข้อมือซึ่งต่ออยู่กับเฟืองหนอนยกขึ้นและลงได้

3.2.5 ชิ้นส่วนกลบริเวณข้อมือหมุนทวน-ตามเข็มนาฬิกา

M_5 มอเตอร์กระแสตรงถูกยึดกับเพลาข้อมือยกขึ้น-ลง ดังนั้นการทำงานของ M_4 กับ M_5 จะให้ผลการเคลื่อนที่ในทิศตั้งฉากกัน

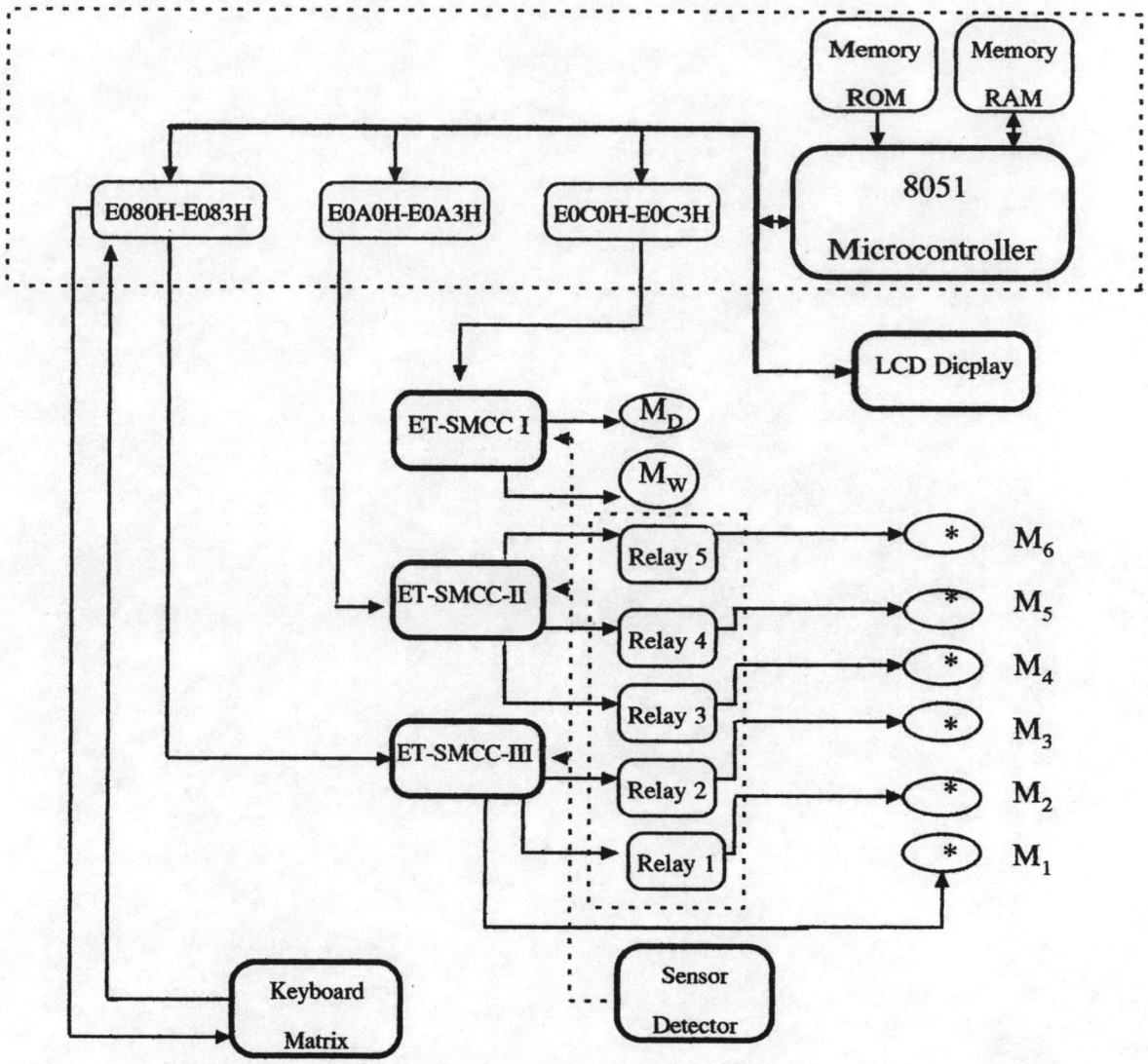
3.2.6 ชิ้นส่วนกลบริเวณมือจับ

ชิ้นส่วนกลบริเวณมือจับจัดในรูปแบบของก้านส่งกำลังขนานกัน ดังในรูปที่ 3.2 ส่วนมือจับเพื่อขับนิ้วมือให้เคลื่อนที่ในลักษณะขนานกับแนวเดิมตลอดเวลา M_6 มอเตอร์กระแสตรงจะส่งกำลังขับผ่านเกลียวหนอน ในการควบคุมการเคลื่อนเข้า-ออกผ่านเกลียวทองเหลือง ซึ่งมีแกนต่อกับก้านส่งกำลังควบคุมการเลื่อนของมือจับ โดยข้อต่อเพลาหมุนชุดมือจับจะส่งแรงผ่านแบริ่งเพื่อให้ระยะว่างของการส่งกำลัง (Clearance) มีค่าน้อยสุด

3.3 การออกแบบระบบควบคุมการเคลื่อนไหวของแขนกล

3.3.1 โครงสร้างของระบบควบคุมด้วยไมโครคอนโทรลเลอร์

การควบคุมการเคลื่อนไหวของแขนกลเลือกใช้ไมโครคอนโทรลเลอร์รุ่น CPAT-32 [10.] ซึ่งใช้ CPU เบอร์ MCS-51 ออกแบบพอร์ตติดต่ออุปกรณ์ภายนอกเป็น 3 พอร์ตผ่านแผ่นวงจร ET-SMCC I, II, III ที่แอดเดรส E080H - E083H, E0A0H - E0A3H และ E0C0H - E0C3H ตามลำดับ คำสั่งควบคุมการเคลื่อนไหวติดต่อผ่านแป้นกดขนาด 25 คีย์และออกแบบให้มีจอ LCD แสดงขั้นตอนของโปรแกรม



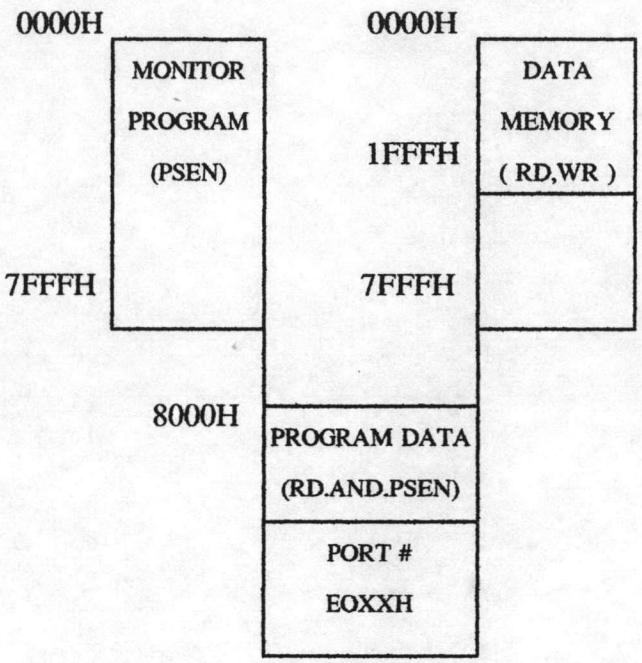
รูปที่ 3.3 แผนภาพแสดงระบบควบคุมการเคลื่อนไหวของแขนกล

การขับมอเตอร์สเต็ปปิงจะขับจากพอร์ตของแผ่นวงจร ET-SMCC [11] ได้โดยตรง แต่การขับมอเตอร์กระแสตรงซึ่งใช้กระแสมากจะขับผ่านรีเลย์ 1 ถึงรีเลย์ 5 และมีการตรวจสอบตำแหน่งทำงานด้วยอุปกรณ์ตรวจจับด้วยแสง เพื่อป้องกันมิให้มีการเคลื่อนที่เกินขีดจำกัด

3.3.2 <u>วงจรเชื่อมโยงสัญญาณ</u>

แผ่นวงจรไมโครคอนโทรลเลอร์จะจัดพอร์ตใช้งานไว้ให้ 3 พอร์ตที่ตำแหน่งแอดเดรส E080H - E083H, E0A0H - E0A3H, E0C0H - E0C3H และจัดหน่วยความจำของสำหรับใช้งานไว้แยกกันระหว่างหน่วยความจำสำหรับข้อมูล (Data Memory) กับหน่วยความจำที่เป็น

โปรแกรมมอนิเตอร์ที่ตำแหน่งแอดเดรส 0000H ถึง 7FFFH การติดต่อกับหน่วยความจำทั้งสองชนิดซึ่งมีแอดเดรสที่เดียวกันนี้จะแยกโดยใช้สัญญาณ PSEN ในการติดต่อกับหน่วยความจำที่เป็นโปรแกรมมอนิเตอร์ และสัญญาณ RD, WR ในการติดต่อกับหน่วยความจำข้อมูล ดังแสดงการจัดหน่วยความจำในรูปที่ 3.4



รูปที่ 3.4 แผนภาพแสดงโครงสร้างของการจัดหน่วยความจำ

ก. การถอดรหัสพอร์ต จะต้องสร้างสัญญาณ IORQ เทียมขึ้นมาโดยการอ้างพอร์ตให้เหมือนหน่วยความจำ ในการติดต่อกับพอร์ตนั้นจะอ้างไปที่หน่วยความจำตำแหน่ง E0000H ทำให้เกิดสัญญาณ IORQ เทียมขึ้น มีผลให้เบอร์พอร์ตถอดรหัสด้วยไบท์ต่ำ ฉะนั้นชื่อพอร์ตที่ CPAT-32 มองเห็นจึงมีชื่อเป็น E0XX ซึ่ง XX ก็คือแอดเดรสไบท์ต่ำนั่นเอง

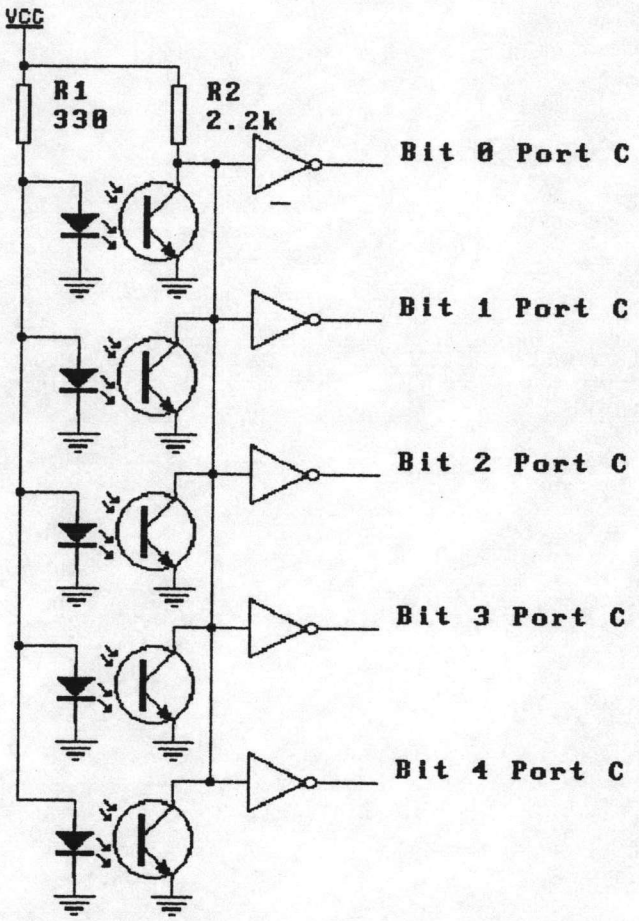
ข. การแสดงผลด้วยจอ LCD ชนิดตัวอักษร จะต้องคำนึงถึงพอร์ตที่จะถูกใช้งาน โดยตำแหน่งของพอร์ต LCD มีรายละเอียดดังนี้

-Base Address อยู่ที่ E060H

-Write Data เขียนตัวอักษรไปที่ LCD เบอร์พอร์ต E062H

-Read Data อ่านข้อมูลที่ตำแหน่ง Cursor อยู่ เบอร์พอร์ตคือ E063H

ค. ชุด Sensor จะใช้ Opto-Sensor MOC71 W1 เป็นตัวตรวจจับสัญญาณจำกัดตำแหน่งในการเคลื่อนที่ของข้อหมุนแต่ละข้อของระบบแขนกลที่ใช้มอเตอร์กระแสตรง ข้อมูลจะส่งเข้าทางพอร์ตทางเข้าเบอร์ 0E0A3H



รูปที่ 3.5 แสดงวงจร Sensor สำหรับระบบแขนกล

3.3.3 <u>วงจรขับชุดมอเตอร์</u>

ชุดขับมอเตอร์เลือกใช้แผ่นวงจร ET-SMCC ซึ่งเป็นแผ่นวงจรที่ออกแบบให้ทำงานกับพอร์ต 8255 บน CPAT-32 แผ่นวงจรET-SMCC 1 ชุดจะใช้ควบคุมมอเตอร์สเต็ปปิงได้ 2 ตัว และรับสัญญาณทางเข้าได้ 5 ช่อง ทำให้สามารถต่อกับชุดตรวจจับสัญญาณเกินขีดจำกัดซึ่งเป็น Opto-Sensor เพื่อป้องกันไม่ให้ข้อหมุนของแขนกลแต่ละข้อเคลื่อนที่เกินขีดจำกัด

3.3.4 วงจรควบคุมทิศทางและตำแหน่งของแขนกล

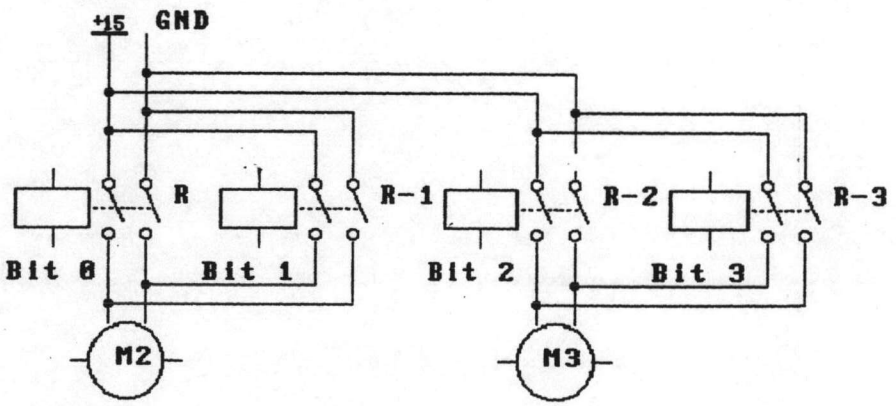
มอเตอร์ต้นกำลังขับเคลื่อน 6 ตัวแบ่งเป็นมอเตอร์สเต็ปปิง 3 ตัวและมอเตอร์กระแสตรง 5 ตัว โดยต้นกำลังขับเคลื่อนชนิดมอเตอร์สเต็ปปิงจะต่อโดยตรงกับ ET-SMCC ด้วยการจัดให้ทำงานแบบจ่ายกระแสไฟพร้อมกันทีเดียว 2 เฟส (Two-Excitation) คือ 0011, 0110, 1100, 1001 หมุนเวียนกันไป ดังตารางการทำงานในรูปที่ 3.6

OUT 4	OUT 3	OUT 2	OUT 1
0	0	1	1
0	1	1	0
1	1	0	0
1	0	0	1

รูปที่ 3.6 แสดงลำดับการจ่ายกระแสไฟชนิด 2 เฟส

ในการกลับทิศทางหมุนทำได้โดยป้อนลำดับการจ่ายกระแสไฟเป็นอีกทิศทางหนึ่งคือ 1001, 1100, 0110, 0011 หมุนเวียนกันไป

สำหรับต้นกำลังขับเคลื่อนซึ่งเป็นมอเตอร์กระแสตรงจะถูกขับผ่านรีเลย์สวิตช์ ซึ่งเป็นชิ้นส่วนที่ทำงานด้วยสนามแม่เหล็กช่วยให้เกิดการตัดต่อและควบคุมทิศทาง การหมุนของมอเตอร์กระแสตรง



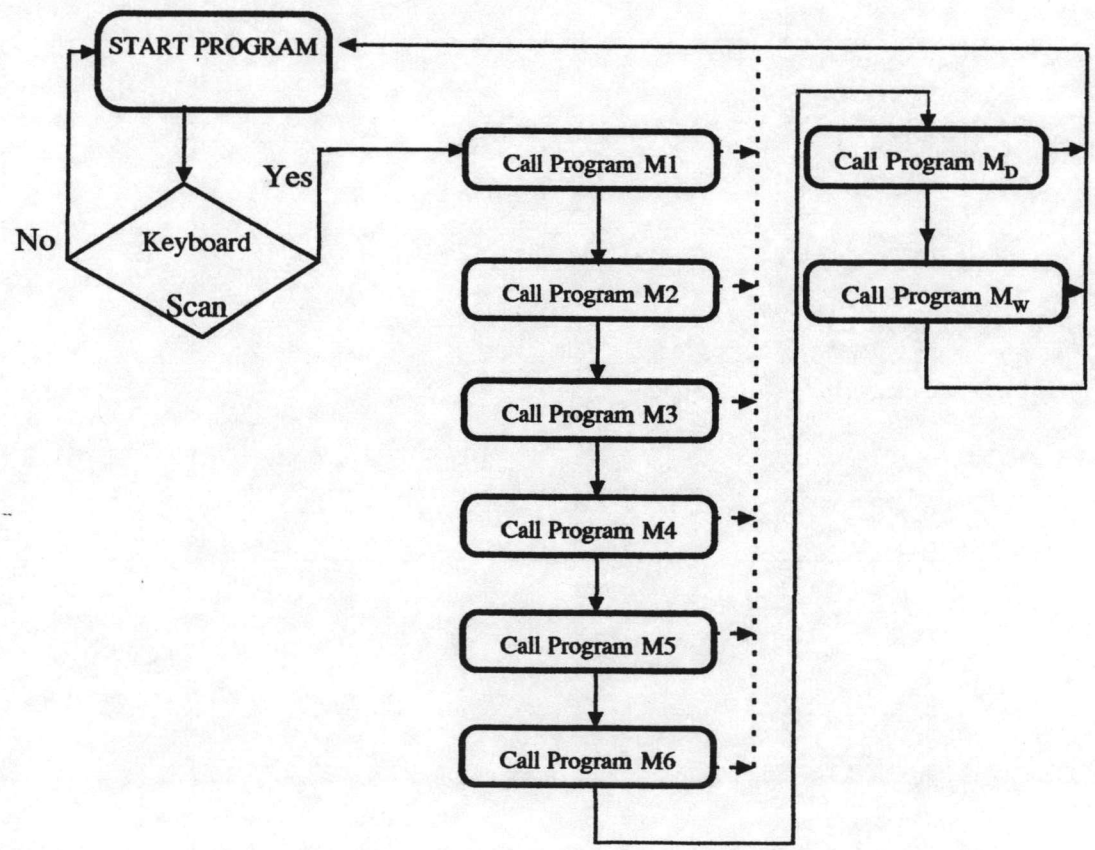
รูปที่ 3.7 วงจรรีเลย์สวิตช์สำหรับควบคุมทิศทางของมอเตอร์กระแสตรง

3.4 โปรแกรมควบคุมการทำงาน

ส่วนของโปรแกรมควบคุมการทำงานของระบบแขนกล ออกแบบให้สามารถควบคุมการติดต่อเพื่อรับ-ส่งข้อมูลทางพอร์ตเข้าและออก โปรแกรมที่ออกแบบเขียนด้วยภาษาเบสิก [12] แบ่งออกได้ 2 ส่วน คือ

ก.1 โปรแกรมควบคุมการทำงานหลักได้แก่ การกำหนดทิศทางการเปลี่ยนแปลงข้อมูลไปยังพอร์ต และควบคุมการรับข้อมูลจากแป้นกด เพื่อควบคุมทิศทางและการทำงานของมอเตอร์

ก.2 โปรแกรมย่อยสนับสนุนการใช้งาน เป็นการรับค่าจากโปรแกรมควบคุมการทำงานหลัก เช่น การกำหนดมอเตอร์ให้หมุนด้วยค่าที่กำหนดขึ้น และการรับสัญญาณตรวจจับขีดจำกัดตำแหน่งการเคลื่อนที่ของข้อต่อ เป็นต้น



รูปที่ 3.8 แผนภาพโครงสร้างของโปรแกรมควบคุมการทำงานของแขนกล

3.4.1 <u>โปรแกรมกำหนดค่าเริ่มต้นของพอร์ตทางเข้าและออก</u>

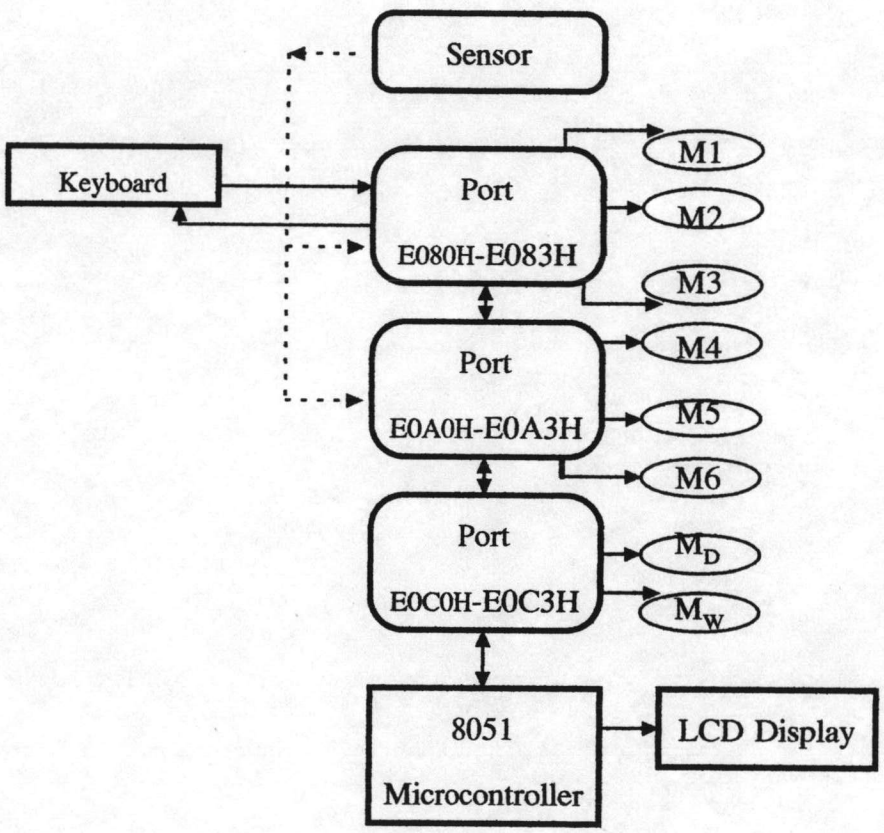
พอร์ตทางเข้าและออกทั้ง 3 ชุดจะถูกนำมาใช้ในการควบคุมการทำงานของแขนกลในส่วนต่างๆ ดังในแผนภาพรูปที่ 3.9 ได้แก่

ก. ควบคุมการขับเคลื่อนมอเตอร์

ข. ควบคุมการรับค่าจาก Keyboard

ค. ควบคุมการแสดงผลออกทางจอ LCD

ง. ควบคุมการทำงานของตัวตรวจจับ



รูปที่ 3.9 แสดงแผนภาพของพอร์ตทางเข้า/ทางออก

ดังนั้นจากแผนภาพสามารถกำหนดค่าโปรแกรมเริ่มต้นได้ ดังนี้

XBY (0E0A3H) = 89H

XBY (0E0C3H) = 82H

XBY (0E083H) = 82H

PA = 0E060H

PB = 0E062H

ค่าของโปรแกรมเริ่มต้นดังกล่าวจะควบคุมพอร์ตต่างๆให้อยู่ในภาวะดังนี้

ก.) เมื่อ XBY (0E0A3H) = 82H จะทำให้

Port A ใช้เป็นพอร์ตทางออกสำหรับส่งสัญญาณควบคุมมอเตอร์ M_4 , M_5 , M_6

Port C ใช้เป็นพอร์ตทางออกสำหรับส่งสัญญาณในแนวแถว (Row) ให้เมตริกซ์คีย์บอร์ด

Port B ใช้เป็นพอร์ตทางเข้าสำหรับรับสัญญาณในแนวคอลัมน์ (Column) จากเมตริกซ์คีย์บอร์ด

ข.) เมื่อ XBY (0E0C3H) = 89H จะทำให้

Port A ใช้เป็นพอร์ตทางออกสำหรับส่งสัญญาณควบคุมมอเตอร์ M_D , M_W

Port C เป็นพอร์ตทางเข้าสำหรับรับสัญญาณจากอุปกรณ์ตรวจจับสัญญาณเกินขีดจำกัด

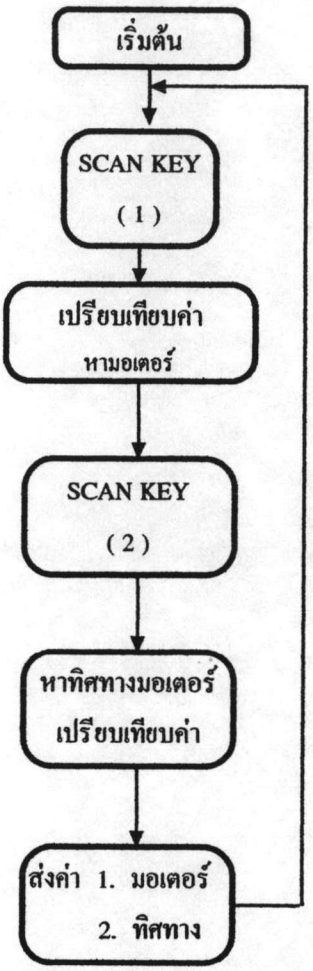
ค.) เมื่อ XBY (0E083H) = 82H จะทำให้

Port A ใช้เป็นพอร์ตทางออกสำหรับส่งสัญญาณควบคุมมอเตอร์ M1, M2, M3

ง.) เมื่อ PA = 0E060H เป็นคำสั่ง Write สำหรับ LCD แบบตัวอักษรใช้ในการเขียนคำสั่งต่างๆ

จ.) เมื่อ PB = 0E062H เป็น Write Data เขียนตัวอักษรไปที่ LCD

3.4.2 โปรแกรมรับข้อมูลจากแป้นควบคุมการขับเคลื่อน




```
10  XBY ( 0E083H ) = 82H
20  XBY ( 0E082H ) = 08H
30  A = XBY ( 0E081H )
40  B = 0FF H
50  C = B AND A
60  IF C = 1 THEN GOTO 70
70  XBY ( 0E082H ) = 10
80  A = XBY ( 0E081H )
90  B = 0FF H
100 C = B.AND.A
110 IF C = 16 THEN GOTO 130 ELSE 120
120 IF C = 4 THEN GOTO 140 ELSE 10
130 REM : MOTOR 1 ทิศทางบวก
140 REM : MOTOR 1 ทิศทางลบ
```

รูปที่ 3.10 แผนผังโปรแกรมรับค่าจากแป้นควบคุมการขับเคลื่อน

1. การทำงานของโปรแกรมรับข้อมูลจากแป้นควบคุมการขับเคลื่อน

1.1 โปรแกรมจะเริ่มต้น โดยการกำหนดค่าเริ่มต้นจากโปรแกรมบรรทัดที่ 10

1.2 การ Scan Key1 : ในชุดแรกเป็นตัวกำหนดมอเตอร์ที่ถูกเลือกให้ทำงานจากโปรแกรมบรรทัด 20-50

1.3 เปรียบเทียบค่า 1 : เป็นคำสั่งเปรียบเทียบระหว่างคีย์ที่ถูกกดกับฐานข้อมูลที่ระบุตัวมอเตอร์จากโปรแกรมบรรทัด 60

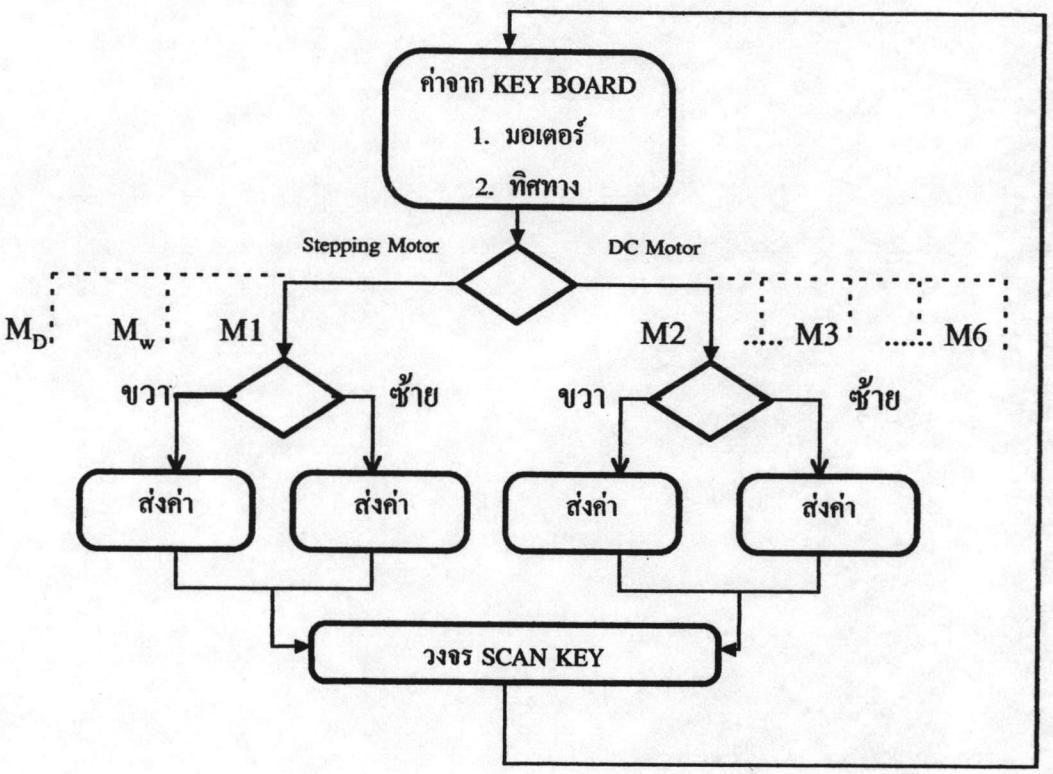
2. การทำงานของโปรแกรมรับค่าเพื่อเลือกทิศทางมอเตอร์

2.1 การ Scan Key 2 : ในชุดสองเป็นตัวกำหนดทิศทางมอเตอร์ที่ถูกเลือกให้ทำงานจากโปรแกรมบรรทัด 70-100

2.2 เปรียบเทียบค่า 2 : เป็นคำสั่งเปรียบเทียบระหว่างคีย์ที่ถูกกดกับฐานข้อมูลเพื่อระบุทิศทางมอเตอร์จากโปรแกรมบรรทัด 110-120

3.4.3 <u>โปรแกรมส่งข้อมูลเพื่อควบคุมมอเตอร์ขับเคลื่อน</u>

ข้อมูลที่ได้จากแป้นกดจะเป็นตัวบอกถึงตำแหน่งของแขนกลที่จะถูกขับเคลื่อนผ่านแผ่นวงจร ET-SMCC



รูปที่ 3.11 แผนผังโปรแกรมส่งข้อมูลเพื่อควบคุมมอเตอร์ขับเคลื่อน

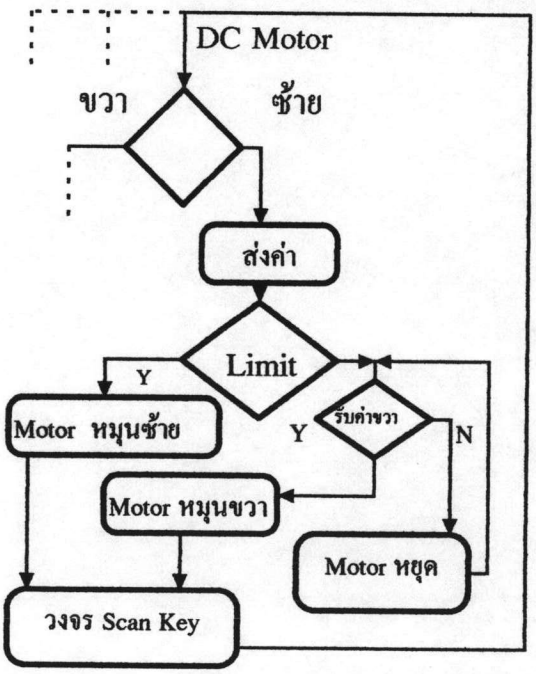
Stepping Motor Control

<u>Right Turn</u>	<u>Left Turn</u>
For A = 1 TO 5	For A = 1 TO 5
XBY (0E080H) = 03H	XBY (0E080H) = 09H
Gosub 6000	Gosub 6000
XBY (0E080H) = 06H	XBY (0E080H) = 0CH

```
Gosub  6000                    Gosub  6000
XBY ( 0E080H ) = 0CH           XBY ( 0E080H ) = 06H
Gosub  6000                    Gosub  6000
XBY ( 0E080H ) = 09H           XBY ( 0E080H ) = 03H
Gosub  6000                    Gosub  6000
Next A                         Next A
```

การทำงานของโปรแกรมส่งข้อมูล เพื่อควบคุมมอเตอร์ขับเคลื่อนชนิดสเต็ปปิงทำงานดังนี้

1. โปรแกรมจะถูกกำหนดเริ่มวัฏจักรการทำงานที่ 1 รอบ โดยการกำหนดให้ For A = 1 To 5
2. จะมีการส่งค่าออกพอร์ต เพื่อเริ่ม Step แรกการหมุนมอเตอร์สเต็ปปิงออกพอร์ต 0E080H ด้วยค่า 03H
3. กระบวนการหน่วงเวลาเพื่อให้ที่พอร์ต 0E080 คงสถานะไว้ด้วยการเรียกโปรแกรมย่อย Gosub 6000
4. ค่าต่อไปถูกส่งออกพอร์ต เพื่อเริ่ม Step ที่สองการหมุนมอเตอร์สเต็ปปิงออกพอร์ต 0E080H ด้วยค่า 06H
5. จากนั้นจะเริ่มกระบวนการหน่วงเวลา โดยจะมีการหน่วงเวลาทุกครั้งที่มีการส่งค่าออกพอร์ต




```
Y = XBY ( 0E0A2H )
Z = 10H
U = Z AND Y
IF U = 16 THEN GOTO @ ELSE #
#...XBY ( 0E080H ) = 10H
GOTO SCAN KEY
@...XBY ( 0E080H ) = 00H
XBY ( 0E082H ) = 10H
A = XBY ( 0E081H )
B = 0FF H
C = B AND A
```

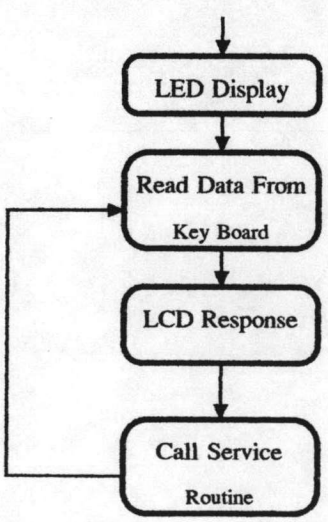
รูปที่ 3.12 แผนผังโปรแกรมควบคุมมอเตอร์กระแสตรง

```
IF C = 16 THEN GOTO ^ ELSE v
XBY ( 0E080H ) = 20H
GOTO @
Y = XBY ( 0E0A2H )
Z = 10H
U = Z AND Y
IF U = 16 THEN GOTO @ ELSE $
```

การทำงานของโปรแกรมส่งข้อมูลเพื่อควบคุมมอเตอร์ขับเคลื่อนชนิดกระแสตรง

1. มอเตอร์ถูกสั่งให้ทำงาน โดยคำสั่ง XBY (0E080H) = 10H
2. จากนั้นโปรแกรมจะทำการตรวจสอบค่ากับ Sensor ก่อนว่ามีมอเตอร์ตัวใดหมุนเกินขีดจำกัดหรือไม่ จากคำสั่ง If U = 16 Then Goto @Else #
3. ถ้ามี Sensor ตรวจพบการเลยขีดจำกัดของมอเตอร์ โปรแกรม Scan จะกระโดดไปที่การ Scan ค่าตรงข้ามการหมุนมอเตอร์ โดยขณะนั้นมอเตอร์จะถูกสั่ง XBY (0E080H) = 00H เพื่อไม่ให้มอเตอร์นั้นทำงาน
4. เมื่อการ Scan ค่าตรงข้ามการหมุนมอเตอร์ ตรวจพบโปรแกรมจะสั่งให้มอเตอร์หมุนในทิศทางตรงข้ามด้วยคำสั่ง XBY (0E080H) = 20H

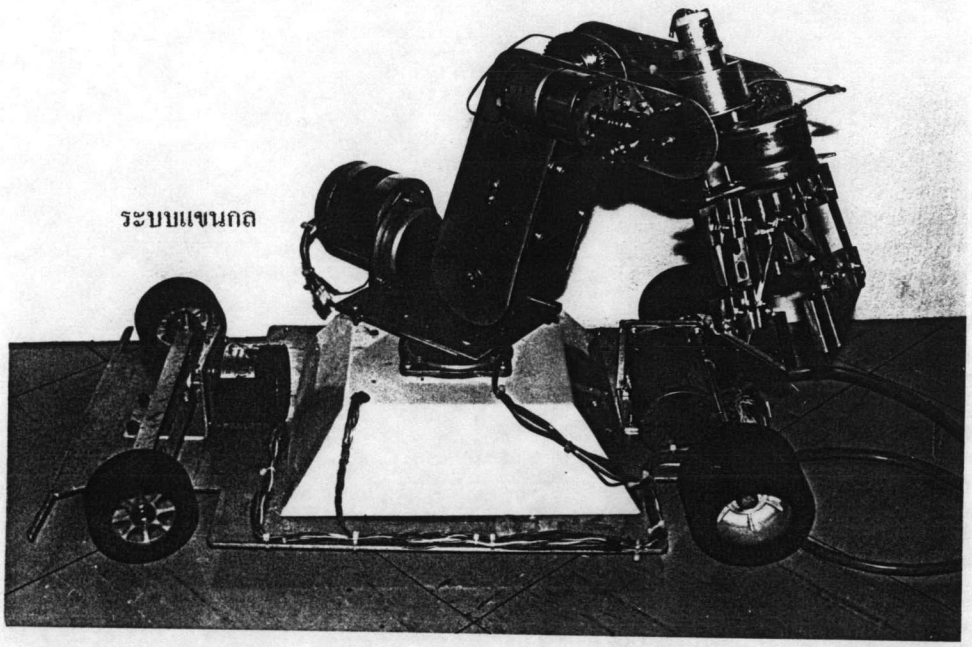
3.4.4 โปรแกรมแสดงผลบนจอ LCD



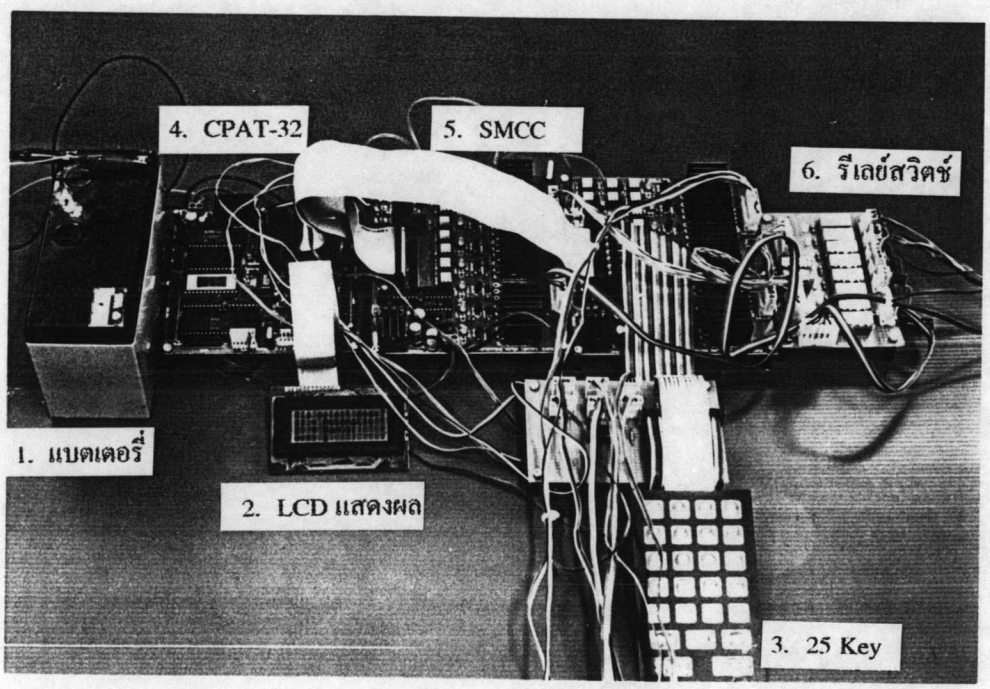

```
STRING 100 , 16
XBY ( PA ) = 56
XBY ( PA ) = 12
$ ( 1 ) = "NUCLEAR"
FOR I = 1 TO 7
XBY ( PA ) = ASC ( $(1) , I )
NEXT
```

รูปที่ 3.13 แผนผังโปรแกรมแสดงผลบนจอ LCD

1. หน้าจอจะถูก Setup ด้วย XBY (PA) = 56 และ XBY (PA) = 12 เพื่อเตรียมรับข้อมูลที่จะถูกแสดงออกหน้าจอ
2. ข้อมูลที่จะถูกแสดงออกหน้าจอ ถูกกำหนดโดยคำสั่ง $ (1) = "Nuclear"
3. โปรแกรมจะส่งทีละตัวอักษรออกด้วยคำสั่ง For I = 1 To 7
4. ข้อมูลถูกส่งออกไปยังหน้าจอทางพอร์ต ด้วยคำสั่ง XBY (PB) = ASC($(1) , I)



รูปที่ 3.14 แสดงแขนกลและระบบควบคุม



รูปที่ 3.15 แสดงระบบควบคุมแขนกล