สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

กล้องติดตามวัตถุ

CAMERA SYSTEM FOR TRACKING A MOVING OBJECT



โดย

นายมนตรี อินตาคำ

นายเมธี บูรณ์จินดา

อาจารย์ที่ปรึกษา

อาจารย์สมเกียรติ วังศิริพิทักษ์

อาจารย์สมศักดิ์ วลัยรัชต์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....
เลขทะเบียน 49985
วัน,เดือน,ปี 16 เม.ย. 2547

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2545

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

เรื่อง กล้องติดตามวัตถุ

CAMERA SYSTEM FOR TRACKING A MOVING OBJECT

ผู้จัดทำ

1. นายมนตรี อินตาคำ รหัสประจำตัว 43015378
2. นายเมธี บูรณ์จินดา รหัสประจำตัว 43015380

_____ อาจารย์ที่ปรึกษา

(อาจารย์สมเกียรติ วังศิริพิทักษ์)

_____ อาจารย์ที่ปรึกษาร่วม

(อาจารย์สมศักดิ์ วลัยรัชต์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กล้องติดตามวัตถุ

นายมนตรี อินตาคำ 43015378
นายเมธี บูรณ์จินดา 43015380
อาจารย์สมเกียรติ วังศิริพิทักษ์ อาจารย์ที่ปรึกษา
อาจารย์สมศักดิ์ วลัยรัชต์ อาจารย์ที่ปรึกษาร่วม

บทคัดย่อ

ในงานบางประเภท งานที่มีความเสี่ยงและเป็นอันตรายต่อมนุษย์ หุ่นยนต์เป็นทางเลือกหนึ่งที่มนุษย์เลือกนำเอามาทำงานแทนมนุษย์ โครงงานกล้องติดตามวัตถุ เป็นโครงงานที่เป็นพื้นฐานระบบการมองของหุ่นยนต์ ซึ่งพื้นฐานระบบการมองนี้ประกอบด้วย กล้องที่ทำหน้าที่รับภาพเข้ามา จากนั้นนำภาพที่ได้ มาทำการประมวลผล ตรวจสอบวัตถุที่เคลื่อนไหว เพื่อติดตามวัตถุที่สนใจนั้น

โครงงานนี้ใช้การนำภาพมาประมวลผล เพื่อประมาณการเคลื่อนที่ของวัตถุ โดยอาศัยหลักวิธี Block-Matching Algorithm คือจะหาส่วนที่เข้ากันได้ (Match)ของมาโครบล็อค (Macroblock) ซึ่งอาจจะมีขนาด 16 x 16 พิกเซล โดยเปรียบเทียบมาโครบล็อกของเฟรมภาพปัจจุบัน กับเฟรมภาพอ้างอิงซึ่งอาจจะเป็นเฟรมก่อนหน้า ในการเปรียบเทียบจะเข้าไปค้นหาในพื้นที่ที่เรียกว่า พื้นที่ค้นหา (Search area) เมื่อเจอส่วนที่เข้ากันได้แล้ว ระยะและทิศทางจากตำแหน่งของมาโครบล็อกในเฟรมอ้างอิงไปยังตำแหน่งของมาโครบล็อกในเฟรมปัจจุบันคือ เวกเตอร์ของการเคลื่อนที่ของวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Camera System for Tracking a Moving Object

Mr. Montree Intakum 43015378

Mr. Metee Boonjinda 43015380

Mr. Somkiat Wangsiripitak Advisor

Mr. Somsak Walairacht Co-Advisor

ABSTRACT

Some kinds of work are too risk and harmful for human beings. Robot is one of the way that is chosen to work for us. Camera System for Tracking a Moving Object is the basis of robot vision which consists of a camera to capture pictures and a system to calculate the vector of the object movement. That system controls a camera to follow the moving object as well.

This project uses Block-matching Algorithm to process and find the matching between macroblock of current frame and reference frame. The macroblock may be 16x16 pixel size, and the reference frame could be the previous frame. Matching is done only in a search area to reduce total search time. When the matching is done completely, the distance and direction from macroblock of reference frame to macroblock of current frame is the vector of the object movement.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานกล้องติดตามวัตถุได้จัดทำขึ้น และสำเร็จลุล่วงไปได้ด้วยดี ในระหว่างการจัดทำโครงงานได้เกิดปัญหาและอุปสรรคมากมาย ซึ่งปัญหาและอุปสรรคเหล่านั้นก็ถูกแก้ไขจนสำเร็จได้ด้วยดี ทั้งนี้เนื่องมาจากผู้จัดทำโครงการได้คำปรึกษาที่ดีจากอาจารย์ที่ปรึกษาซึ่งก็คือ อาจารย์สมเกียรติ วังศิริพิทักษ์ และอาจารย์สมศักดิ์ วลัยรัชต์ ที่ยินดีให้คำปรึกษา และให้คำแนะนำที่ดีทุกเรื่องตลอดมา ซึ่งต้องขอบพระคุณอาจารย์เป็นอย่างมาก อีกทั้งรุ่นพี่และเพื่อนๆในภาควิชาวิศวกรรมคอมพิวเตอร์ของเราก็มีส่วนช่วยให้โครงงานนี้สำเร็จด้วยดี

และต้องขอบพระคุณบุคคลที่สำคัญยิ่ง ที่ทำให้ผู้จัดทำมีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้จัดทำเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่เสมอมา ในทุกๆด้านอันหาที่เปรียบมิได้ ผู้จัดทำขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

มนตรี อินตาคำ

เมธี บรูณ์จินดา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VI
สารบัญภาพ	VII
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของงานวิจัย	1
1.3 ขอบเขตของงานวิจัย	1
1.4 วิธีการดำเนินงาน	2
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 ไมโครคอนโทรลเลอร์ MCS-51	3
2.2 สเต็ปปิ้งมอเตอร์ (Stepping Motor)	15
2.3 การประมวลผลภาพเพื่อทำการตรวจสอบการเคลื่อนที่ของวัตถุ	20
บทที่ 3 โครงสร้างและการออกแบบ	24
3.1 ภาพรวมของระบบ	24
3.2 การออกแบบโครงสร้างของฐานกล้อง	24
3.3 การออกแบบวงจรควบคุมการทำงาน	26
3.4 การออกแบบโปรแกรมควบคุมสเต็ปปิ้งมอเตอร์ของไมโครคอนโทรลเลอร์ MCS-51	28
3.5 การออกแบบหน้าจอ	30
3.6 คลาสไดอะแกรม	31
3.7 ลักษณะของคลาสต่างๆ	31
บทที่ 4 ผลการทดลอง/การทดสอบ	
4.1 หาความเร็วเฉลี่ยของสเต็ปปิ้งมอเตอร์	34
4.2 หาอัตราส่วนการเคลื่อนที่ของวัตถุกับจำนวนสเต็ปปิ้งมอเตอร์	35
4.3 หาเวลาเฉลี่ยที่ใช้ในการประมวลผลภาพและสั่งควบคุมกล้อง	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
บทที่ 5 บทวิจารณ์และสรุป	39
5.1 บทวิจารณ์	39
5.2 สรุป	39
5.3 ข้อเสนอแนะสำหรับการพัฒนาในอนาคต	40
บรรณานุกรม	41



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงหน้าที่ของขาต่าง ๆ ในพอร์ต 3	5
2.2 แสดงการเลือกอัตราบอดของวงจรพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ MCS-51	11
2.3 แสดงการควบคุมสเต็ปปิ้งมอเตอร์แบบหนึ่งเฟสหรือแบบฟูลสเต็ป(Full step)	17
2.4 แสดงการควบคุมสเต็ปปิ้งมอเตอร์แบบ 2 เฟส	17
2.5 แสดงการควบคุมสเต็ปปิ้งมอเตอร์แบบครึ่งสเต็ป	18
4.1 แสดงผลจากการวัดความเร็วในการหมุนซ้าย-ขวา โดยรูปแบบของเวลาคือ นาที:วินาที:หนึ่งส่วนร้อยวินาที	34
4.2 แสดงผลจากการวัดความเร็วในการหมุนขึ้น-ลง โดยรูปแบบของเวลาคือ นาที:วินาที:หนึ่งส่วนร้อยวินาที	35
4.3 แสดงผลการบันทึกจากการทดลอง หาอัตราส่วนการเคลื่อนที่ในแนวซ้าย-ขวา	35
4.4 แสดงผลการบันทึกจากการทดลอง หาอัตราส่วนการเคลื่อนที่ในแนวขึ้น-ลง	36
4.5 แสดงผลการบันทึกจากการทดลอง การหาเวลาที่ใช้ในการประมวลผลภาพและการสั่งการควบคุมกล้อง	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่ หน้า

2.1 แสดงโครงสร้างภายในของ MCS-51 4
2.2 แสดงขาต่าง ๆ ของไมโครคอนโทรลเลอร์ MCS-51 5
2.3 แสดงโครงสร้างพอร์ตทั้ง 4 ของ MCS-51 7
2.4 แสดงการต่อพอร์ตเข้ากับระบบบัสภายในของ MCS-51 7
2.5 แสดงรูปแบบข้อมูลอนุกรมแบบอะซิงโครนัส 8
2.6 แสดงรีจิสเตอร์ควบคุมการทำงานของพอร์ตอนุกรม 10
2.7 แสดงวงจรเชื่อมต่อ MAX232 หรือ ICL232 เข้ากับพอร์ตอนุกรม ของคอมพิวเตอร์และไมโครคอนโทรลเลอร์ 14
2.8 แสดงสเต็ปปิ้งมอเตอร์ แบบมีสาย 5 เส้น 15
2.9 แสดงมอเตอร์แบบมีสาย 6 เส้น 15
2.10 แสดงโครงสร้างของสเต็ปปิ้งมอเตอร์ 16
2.11 แสดง (ก) โครงสร้าง (ข) วงจรเทียบเท่า (equivalent circuit) ของมอเตอร์ ชนิด 4 ขด 16
2.12 แสดงสเต็ปปิ้งมอเตอร์ ชนิดมีสาย 6 เส้น 18
2.13 แสดงสเต็ปปิ้งมอเตอร์ ชนิดมีสาย 5 เส้น 19
2.14 แสดงการใช้มาโครบล็อคจากเฟรมปัจจุบันเข้าไปค้นหาส่วนที่เข้ากันได้ในวินโดว์ค้นหา ภายในเฟรมอ้างอิง จนเจอส่วนที่ เข้ากันได้แล้วก็จะได้เวกเตอร์ของการเคลื่อนที่ 20
2.15 แสดงเฟรม Ft-2 21
2.16 แสดงเฟรม Ft-1 21
2.17 แสดงเฟรม Ft 21
2.18 แสดงสี่เหลี่ยมครอบบริเวณที่ไม่เหมือนกันของเฟรม Ft-2 กับ Ft-1 22
2.19 แสดงสี่เหลี่ยมครอบบริเวณที่ไม่เหมือนกันของเฟรม Ft-1 กับ Ft 22
2.20 แสดงสี่เหลี่ยมครอบส่วนที่ทับกัน ดังเส้นสีน้ำเงิน 22
2.21 แสดงการนำสี่เหลี่ยมที่ได้จากรูปที่ 2.20 มาวางในเฟรม Ft ซึ่งก็จะได้ขนาดและตำแหน่ง ของวัตถุที่เคลื่อนที่อยู่ 23
2.22 แสดงเวกเตอร์ที่ใช้เป็นข้อมูลในการบังคับอุปกรณ์ให้ทำการหมุนซ้าย-ขวาและก้ม-เงย 23
3.1 แสดงภาพรวมของโครงงานกล้องติดตามวัตถุ 24
3.2 แสดงโครงสร้างของฐานกล้อง 25
3.3 แสดงส่วนไมโครคอนโทรลเลอร์ที่เชื่อมต่อกับวงจรบัฟเฟอร์และพอร์ตอนุกรมของ คอมพิวเตอร์ 26
3.4 แสดงวงจรไดรเวอร์สเต็ปปิ้งมอเตอร์ 28
3.5 แสดง Flowchart ของโปรแกรมควบคุมสเต็ปปิ้งมอเตอร์ 29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

รูปที่		หน้า
3.6	แสดงหน้าจอโปรแกรมของระบบ	30
3.7	แสดงคลาสไดอะแกรมของโปรแกรม	31
3.8	แสดงลักษณะของ คลาส CSerial	32
3.9	แสดงลักษณะของคลาส CControlHardware	32
3.10	แสดงลักษณะของคลาส CSampleGrabberCB	32
3.11	แสดงลักษณะของคลาส CImgProcess	33
3.12	แสดงลักษณะบางส่วนของคลาส CStillCapDlg	33

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

ในปัจจุบันงานบางอย่างที่มนุษย์ไม่สามารถที่จะปฏิบัติได้ เนื่องจากอาจได้รับความเสี่ยงต่อสุขภาพ จึงต้องมีการคิดค้นและสร้างเครื่องจักรกลที่มาทำงานที่มีความเสี่ยงเช่นนี้ หุ่นยนต์เป็นจักรกลชิ้นหนึ่งซึ่งเป็นสิ่งประดิษฐ์ที่มนุษย์สร้างขึ้นมาทำงานแทนมนุษย์ โครงงานกล้องติดตามวัตถุ เป็นโครงงานที่ทำขึ้นเพื่อเป็นพื้นฐานทางระบบการมองเห็นของหุ่นยนต์ โดยระบบการมองนี้ จะใช้กล้องเป็นตัวรับภาพเข้ามาแล้วมีซอฟต์แวร์ ที่ทำหน้าที่ตรวจสอบการเคลื่อนที่ของวัตถุ และบังคับให้อุปกรณ์บังคับการเคลื่อนที่ของกล้องในลักษณะการ หันซ้าย-ขวา (pan) และ ก้ม-เงย (tilt) ตามการเคลื่อนที่ของวัตถุ เพื่อให้วัตถุอยู่ในมุมมองของกล้องตลอดเวลา ผลที่คาดว่าจะได้รับจากโครงงานนี้ก็คือสามารถควบคุมให้กล้องติดตามวัตถุที่สนใจ และสามารถนำไปประยุกต์เป็นระบบการมองเห็นของหุ่นยนต์ได้

1.2 วัตถุประสงค์ของงานวิจัย

สร้างพื้นฐานระบบการมองเห็นของหุ่นยนต์ โดยใช้กล้องวีดิโอเป็นตัวรับภาพเข้ามาแล้วทำการประมวลผลภาพที่รับเข้ามา เพื่อตรวจสอบการเคลื่อนที่ของวัตถุ แล้วบังคับสั่งงานอุปกรณ์บังคับการเคลื่อนที่ของกล้องให้หมุนซ้าย-ขวา และก้ม-เงย ตามการเคลื่อนที่ของวัตถุที่สนใจ เพื่อให้วัตถุอยู่ในมุมมองของกล้องตลอดเวลา

1.3 ขอบเขตของงานวิจัย

ระบบกล้องติดตามวัตถุประกอบด้วย 2 ส่วนหลัก คือ ส่วนฮาร์ดแวร์ (Hardware) และส่วนซอฟต์แวร์ (Software)

1. ส่วนฮาร์ดแวร์ ประกอบด้วยกล้องถ่ายภาพวีดิโอ 1 ตัวที่ทำหน้าที่รับภาพเข้ามา และระบบควบคุมบังคับการเคลื่อนที่ของกล้องในทิศทางหมุนซ้าย-ขวา และก้ม-เงย ตามการเคลื่อนที่ของวัตถุ

2. ส่วนซอฟแวร์ เป็นส่วนของโปรแกรมที่ติดตั้งอยู่บนเครื่องคอมพิวเตอร์ จะรับภาพจากกล้องมาประมวลผล และตรวจสอบการเคลื่อนที่ของวัตถุ แล้วบังคับให้ระบบควบคุมบังคับการเคลื่อนที่ของกล้องทำการหมุนซ้าย-ขวา และก้ม-เงย ตามการเคลื่อนที่ของวัตถุ เพื่อให้วัตถุอยู่ในมุมมองของกล้องตลอดเวลา โดยจำกัดวัตถุที่เคลื่อนที่ ที่สนใจและติดตามเพียง 1 ชิ้น โปรแกรมควบคุมระบบกล้องติดตามวัตถุสามารถแสดงผล และสามารถควบคุมการหันซ้าย-ขวาและ ก้ม-เงย 2 โหมดด้วยกัน คือให้บังคับโดยผู้ใช้ (Manual Mode) และทำการหันซ้าย-ขวาและ ก้ม-เงย ตามการเคลื่อนที่ของวัตถุโดยอัตโนมัติ (Auto Mode)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 วิธีการดำเนินงาน

เริ่มจากการศึกษาทฤษฎีพื้นฐานต่างๆ ที่เกี่ยวข้อง ได้แก่ ศึกษาและหาข้อมูลในการจัดการกับไฟล์รูปภาพ, ศึกษาการดึงภาพจากกล้อง, ศึกษาแนวทางการตรวจสอบการเคลื่อนที่ของวัตถุ เพื่อใช้ในการประมวลผลภาพ, ศึกษาการทำงานของไมโครคอนโทรลเลอร์ MCS-51, ศึกษาการเขียนโปรแกรมคอนโทรล (Control) ตัว MCS-51 เพื่อควบคุม สเต็ปปิ้งมอเตอร์ (Stepping Motors) และเพื่อติดต่อรับส่งข้อมูลกับคอมพิวเตอร์ผ่านคอมพอร์ต (RS232)

หลังจากที่ได้ศึกษาทฤษฎีต่างๆ แล้วก็นำทฤษฎีเหล่านั้นมาใช้งาน ในส่วนของซอฟต์แวร์ก็ทำการเขียนโปรแกรมดึงภาพจากกล้อง, เขียนโปรแกรมจัดการกับไฟล์รูปภาพ, เขียนโปรแกรมตรวจสอบการเคลื่อนที่ของวัตถุ โดยใช้ทฤษฎีที่ได้ศึกษามา, เขียนโปรแกรมคอนโทรลตัว MCS-51 เพื่อควบคุมสเต็ปปิ้งมอเตอร์ และเพื่อติดต่อรับส่งข้อมูลกับคอมพิวเตอร์ผ่านคอมพอร์ต และในส่วนของ ฮาร์ดแวร์ทำการออกแบบระบบควบคุมบังคับการเคลื่อนที่ของกล้องในทิศทางการหันซ้าย-ขวา และ ก้ม-เงย, ออกแบบวงจรส่วนประมวลผล MCS-51, ออกแบบวงจรขับสเต็ปปิ้งมอเตอร์ และวงจรเพาเวอร์ซัพพลาย (Power Supply) แล้วลงมือทำในส่วนฮาร์ดแวร์

เมื่อลงมือทำระบบทั้งส่วนฮาร์ดแวร์ และซอฟต์แวร์ เสร็จแล้วก็ทำการทดสอบระบบ แก้ไขจุดบกพร่องที่พบจากการทดสอบ และเพิ่มประสิทธิภาพให้แก่ระบบ ท้ายสุดก็นำผลการทดลองมาทำการวิจารณ์และสรุป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

2.1 ไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ (Microcontroller) เป็นชื่อของอุปกรณ์อิเล็กทรอนิกส์แบบหนึ่งที่บรรจุความสามารถมากมายไม่ว่าจะเป็นหน่วยประมวลผล หน่วยคำนวณทางคณิตศาสตร์และลอจิก วงจรรับสัญญาณอินพุต วงจรขับสัญญาณทางเอาต์พุต หน่วยความจำ วงจรกำเนิดสัญญาณนาฬิกา ทำให้ไมโครคอนโทรลเลอร์สามารถนำไปประยุกต์ใช้งานแทนวงจรอิเล็กทรอนิกส์ที่ซับซ้อนได้เป็นอย่างดี โดยช่วยลดจำนวนของอุปกรณ์และขนาดของระบบลง ในขณะที่มีขีดความสามารถสูงขึ้นภายใต้งบประมาณที่เหมาะสม

ไมโครคอนโทรลเลอร์มาจากคำสองคำคือ "ไมโคร" (Micro) ซึ่งหมายถึงไมโครโปรเซสเซอร์ (Microprocessor) ซึ่งเป็นอุปกรณ์ประมวลผลข้อมูลขนาดเล็ก ซึ่งภายในประกอบด้วยหน่วยประมวลผลกลางหรือซีพียู (CPU : Central Processing Unit) หน่วยคำนวณทางคณิตศาสตร์และลอจิก (ALU : Arithmatic Logic Unit) วงจรเชื่อมต่อหน่วยความจำ และวงจรเชื่อมต่อสัญญาณนาฬิกา อีกคำหนึ่งคือ คำว่า "คอนโทรลเลอร์"(Controller) หมายถึงอุปกรณ์ควบคุม ดังนั้นไมโครคอนโทรลเลอร์จึงเป็นอุปกรณ์ที่ใช้ในการควบคุม โดยที่สามารถเขียนโปรแกรมเพื่อกำหนดรูปแบบการควบคุมได้อย่างอิสระ

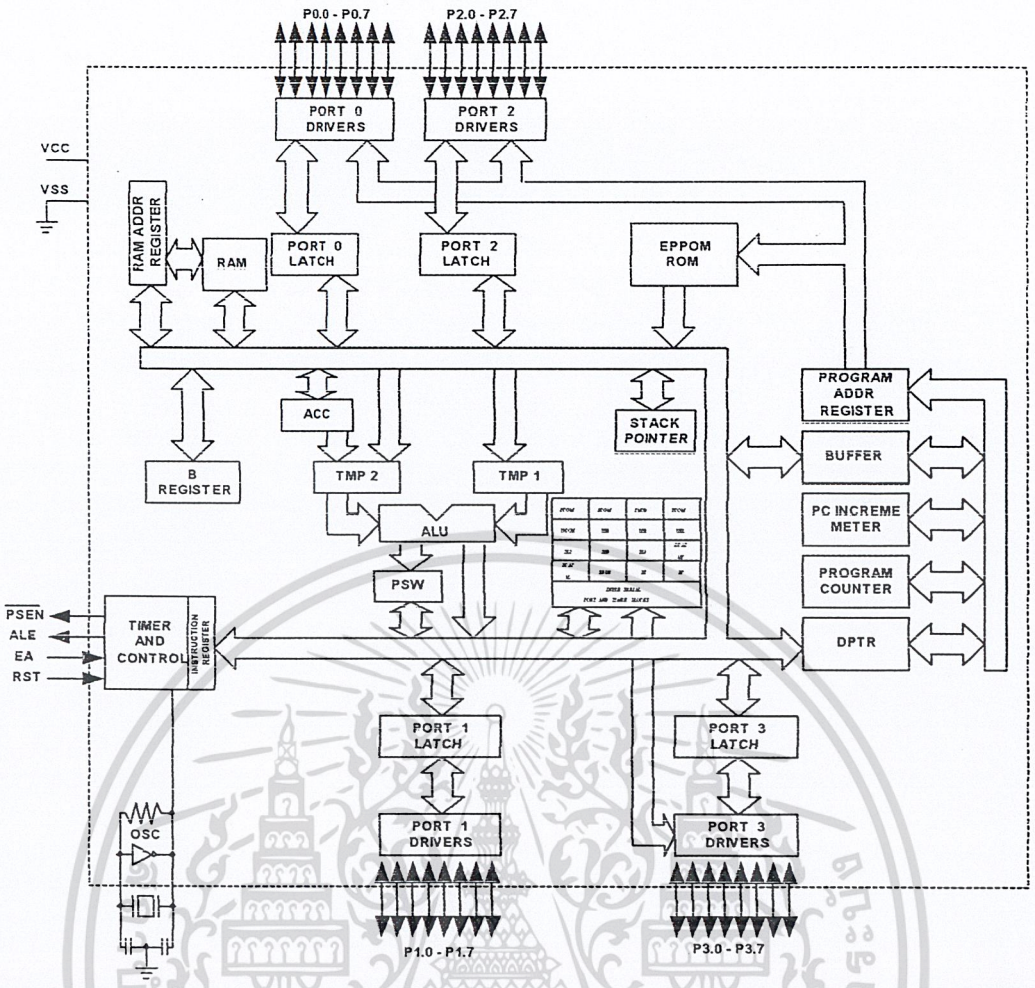
2.1.1 คุณสมบัติทางเทคนิคของไมโครคอนโทรลเลอร์ ตระกูล MCS-51

- เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูขนาด 8 บิต
- ภายในมีหน่วยความจำโปรแกรมเป็นแบบแฟลชสามารถลบและเขียนใหม่ได้
- หน่วยความจำข้อมูลพื้นฐานเป็นหน่วยความจำแบบแรม ในบางเบอร์จะมีหน่วยความจำแบบอีอีพรอมเพิ่มเติม
- ขาพอร์ตเป็นแบบ 2 ทิศทาง สามารถใช้งานได้เป็นทั้งอินพุตและเอาต์พุต
- มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์
- ไทเมอร์/เคาน์เตอร์ขนาด 16 บิต อย่างน้อย 2 ตัว
- สามารถรองรับแหล่งกำเนิดอินเตอร์รัปต์ได้ 6 ประเภท
- สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์
- มีวงจรกำเนิดสัญญาณนาฬิกาภายในชิป
- มีวงจรสื่อสารอนุกรมแบบ SPI
- มีวอตช์ด็อกไทเมอร์ในตัว

2.1.2 โครงสร้างของ MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีด้วยกันหลายเบอร์ขึ้นกับโครงสร้างภายใน บางเบอร์จะมีหน่วยความจำภายในเป็นแบบ ROM บางเบอร์เป็นแบบ EPROM บางเบอร์มี RAM ภายใน 128 ไบต์ บางเบอร์มี 256 ไบต์ เป็นต้น รายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51 แสดงดังรูปที่ 2.1

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1แสดงโครงสร้างภายในของ MCS-51

2.1.3 การจัดขาต่าง ๆ ของ MCS-51

ไอซี ไมโครคอนโทรลเลอร์ MCS-51 โครงสร้าง IC เป็นแบบ DIP มีขาทั้งหมด 40 ขาโดยขาต่าง ๆ จะใช้เป็นขาพอร์ตอินพุต, เอาต์พุต, ขาสัญญาณควบคุม, ขาตำแหน่งหน่วยความจำ และขาข้อมูลดังรูปที่ 2.2

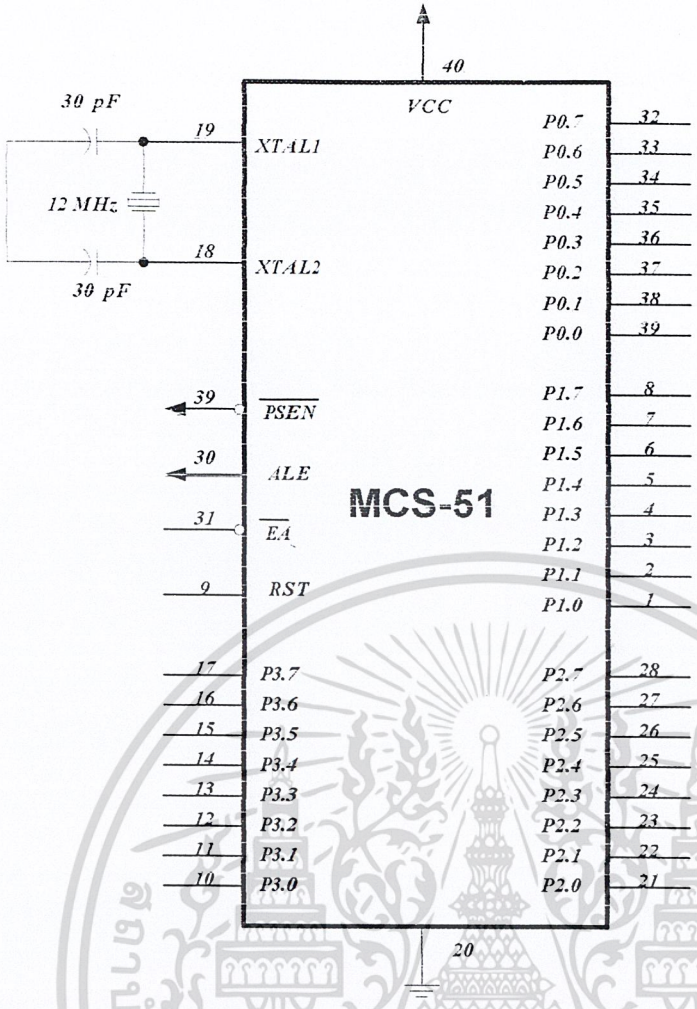
ความหมายของขาต่าง ๆ มีดังนี้

2.1.3.1 พอร์ต 0 **(Port 0)** พอร์ต 0 ได้แก่ขาที่ 32-39 ของ MCS-51 สามารถใช้เป็นอินพุต เอาต์พุตได้ นอกจากนี้ในการติดต่อกับหน่วยความจำภายนอกยังใช้เป็นขา Address Bus และ Data Bus อีกด้วย

2.1.3.2 พอร์ต 1 **(Port 1)** พอร์ต 1 ได้แก่ขาที่ 1-8 เป็นพอร์ต 8 บิต สามารถอ้างทีละบิตได้ คือ P1.0, P1.1,..., P1.7

2.1.3.3พอร์ต 2 **(Port 2)** พอร์ต 2 ได้แก่ขาที่ 21-28 จะใช้งาน 2 หน้าที่ คือใช้เป็นพอร์ต 8 บิตกับใช้เป็นขาแอดเดรส 8 บิตในการอ้างหน่วยความจำภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดงขาต่าง ๆ ของไมโครคอนโทรลเลอร์ MCS-51

2.1.3.4 พอร์ต 3 (Port 3) พอร์ต 3 ได้แก่ขาที่ 10-17 จะใช้งานสองหน้าที่คือ เป็นพอร์ตอินพุตและเอาต์พุต และใช้เป็นขาควบคุมต่าง ๆ ดังตารางที่ 2.1

บิต	ชื่อ	หน้าที่พิเศษ
P3.0	RXD	ใช้รับข้อมูลทางพอร์ทอนุกรม
P3.1	TXD	ใช้ส่งข้อมูลทางพอร์ทอนุกรม
P3.2	INT0	อินเทอร์รัปต์ภายนอกหมายเลข 0
P3.3	INT1	อินเทอร์รัปต์ภายนอกหมายเลข 1
P3.4	T0	ตัวจับเวลา/ตัวนับ ตัวที่ 0
P3.5	T1	ตัวจับเวลา/ตัวนับ ตัวที่ 1
P3.6	$\overline{WR}$	สัญญาณเขียนข้อมูลหน่วยความจำภายนอก
P3.7	$\overline{RD}$	สัญญาณอ่านข้อมูลหน่วยความจำภายนอก

ตารางที่ 2.1 แสดงหน้าที่ของขาต่าง ๆ ในพอร์ต 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3.5 **PSEN (Program Store Enable)** ขา PSEN เป็นขาที่ส่งสัญญาณออก ได้แก่ ขา 29 ขานี้จะแอคทีฟเมื่อ MCS-51 ต้องการอ่าน Code โปรแกรมภายนอก โดยปกติถ้าหน่วยความจำภายนอกเป็น EPROM ขา PSEN จะต่อกับขา Output Enable (OE) ของ EPROM

2.1.3.6 **ALE (Address Latch Enable)** เนื่องจากพอร์ต 0 สามารถใช้เป็นขาอ้างตำแหน่งและขาข้อมูล MCS-51 จึงมีขา ALE ได้แก่ขา 30 Multiplex สัญญาณ Address bus ของ Port 0 ในการใช้งานระบบ MCS-51 นั้นจำเป็นต้องมีอุปกรณ์มาต่อกับ Port 0 ที่ทำหน้าที่ Latch สัญญาณ Address Bus เมื่อ MCS-51 ต้องการติดต่อกับหน่วยความจำภายนอก MCS-51 จะส่งสัญญาณ Address Bus ออกมาก่อนทาง Port 0 จากนั้นจะส่งสัญญาณ ALE มา Latch อุปกรณ์ภายนอก ให้เก็บค่า Address Bus ของ Port 0 ไว้เพื่อใช้ Port เป็น Data Bus ต่อไป

2.1.3.7 **EA (External Access)** EA ได้แก่ขาที่ 31 ถ้าขานี้เป็นลอจิก "1" จะใช้เบอร์ 8051/8052 เพื่อบอกว่าให้อ่านโปรแกรมจากหน่วยความจำโปรแกรมภายใน แต่ถ้าเป็นลอจิก "0" จะบอกว่าให้ MCS-51ทำโปรแกรมโดย อ่านจากหน่วยความจำโปรแกรมภายนอก (ถ้าขา EA เป็น "0" ขา PSEN จะแอคทีฟ) ถ้าหากเป็นเบอร์ 8031 หรือ8032 ขา EA จะเป็น "0" เสมอ เพราะว่าไม่มีโปรแกรมหน่วยความจำภายใน แต่ถ้าใช้เบอร์ 8051/8052 ซึ่งมีหน่วยความจำโปรแกรมภายในและให้ขา EA เป็น "0" ซึ่งจะ Disabled ROM ภายในและจะอ่านโปรแกรมจาก EPROM ภายนอกแทน

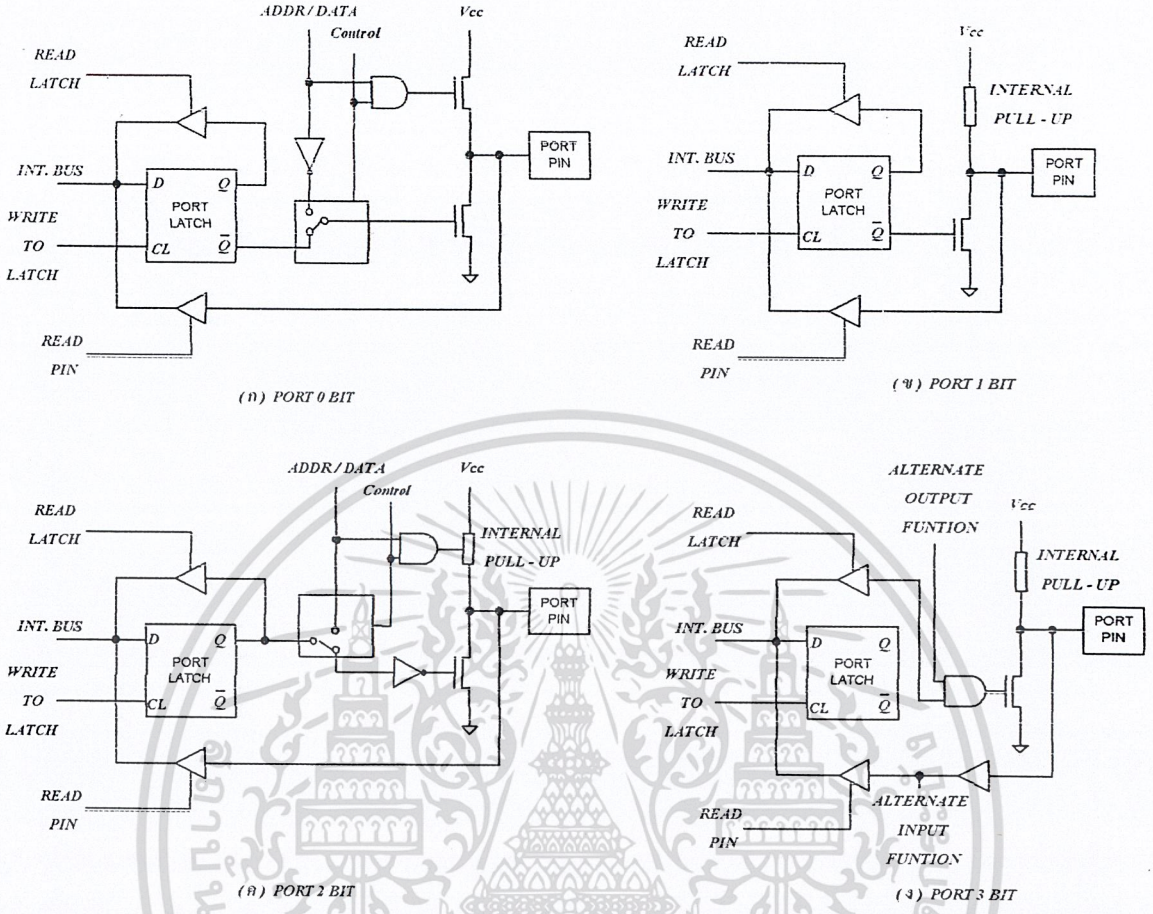
2.1.3.8 **RST (Reset)** ขา RST ได้แก่ขา 9 จะใช้ในการรีเซต MCS-51 โดยจะให้ขานี้เป็นลอจิก "1" อย่างน้อย 2 Machine Cycles จึงจะรีเซตระบบได้

2.1.4 โครงสร้างของพอร์ตอินพุตเอาต์พุต **(I/O Port Structure)**

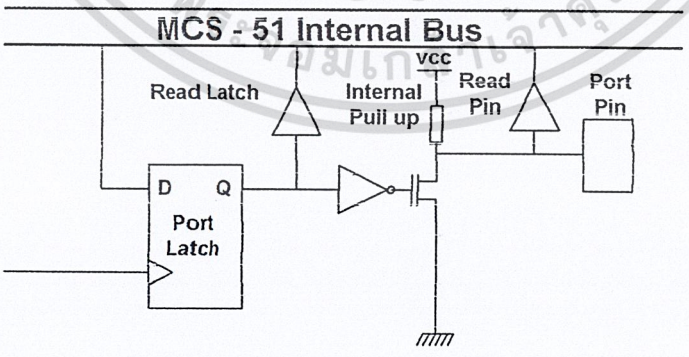
ขาของพอร์ตจะแสดงโครงสร้างภายในได้ ดังรูปที่ 2.3 โดยจะมีโครงสร้างเป็น Field-effect Transistor ต่ออยู่กับขาภายนอก และมีความต้านทานต่อ Pull-up อยู่สำหรับพอร์ต 1, 2, 3 แต่ถ้าเป็นพอร์ต 0 จะไม่มีตัวต้านทาน Pull-up ภายใน เพราะว่าต้องใช้เป็นขา Address Bus และ Data Bus

พอร์ตนี้สามารถใช้เป็นอินพุตเอาต์พุตกับอุปกรณ์ภายนอกได้ ในการอ่านข้อมูลจากพอร์ตจะอ่านได้สองแบบ คือ Read Latch และ Read Pin โดย Read Latch หมายถึงการอ่านข้อมูลที่ถูก Latch เอาไว้ เข้าสู่บัสภายในของ MCS-51 เช่น การทำคำสั่ง CPL P1.5 แต่ถ้าเป็นการ Read Pin จะเป็นการใช้พอร์ตเป็นอินพุต โดยจะอ่านค่าจากขาของไอซีเข้าสู่บัสภายใน การอ่านแบบ Read Latch และ Read Pin จะมีสัญญาณมาควบคุมที่บัฟเฟอร์ดังรูปที่ 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 แสดงโครงสร้างพอร์ตทั้ง 4 ของ MCS-51



รูปที่ 2.4 แสดงการต่อพอร์ตเข้ากับระบบบัสภายในของ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5 พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51

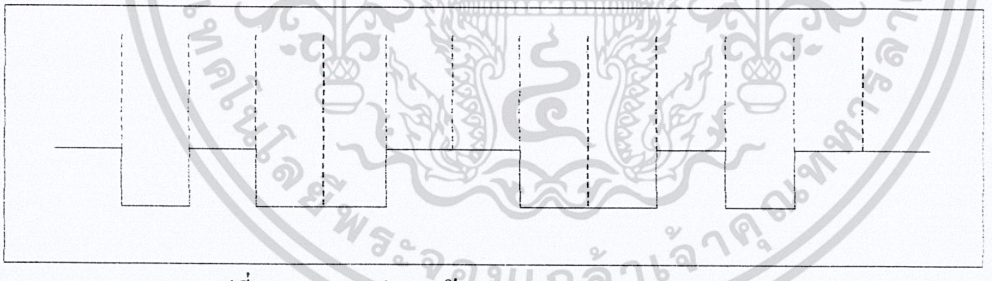
ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์ 1 ชุด (วงจรสื่อสารแบบฟูลดูเพล็กซ์ หมายถึง วงจรสื่อสารที่สามารถทำการรับและส่งข้อมูลในลักษณะ 2 ทิศทาง ได้ในเวลาเดียวกัน) โดยใช้ขาสัญญาณของพอร์ต 3 คือ ขา P3.0 เป็นขารับข้อมูลเข้าหรือ RxD และขาP3.1 เป็นขาส่งข้อมูลออกหรือ TxD โดยวงจรสื่อสารข้อมูลแบบอนุกรมของไมโครคอนโทรลเลอร์ตระกูล MCS-51 แบบแฟลชเป็นแบบอะซิงโครนัส ปกติแล้วพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 จะใช้ในการติดต่อสื่อสารกับพอร์ตอนุกรมของคอมพิวเตอร์ โดยใช้มาตรฐาน RS-232 แต่ในปัจจุบันสามารถติดต่อกันในมาตรฐาน RS-422 หรือ RS-485 ได้แล้ว โดยใช้ไอซีพิเศษที่ทำหน้าที่ในการแปลงสัญญาณการสื่อสารดังกล่าว

2.1.5.1 การสื่อสารข้อมูลแบบอะซิงโครนัส

การสื่อสารข้อมูลแบบอะซิงโครนัสคือการรับและส่งข้อมูลโดยไม่จำเป็นต้องมีสัญญาณนาฬิการ่วมด้วย แต่จะใช้การกำหนดอัตราเร็วในการรับและส่งข้อมูลให้มีค่าเท่ากัน เรียกอัตราเร็วนี้ว่า อัตราบอด หรือ บอดเรต (baud rate)มีหน่วยเป็น บิตต่อวินาที (bit per second : bps)

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนด้วยกันคือ

1. บิตเริ่มต้น (start bit) มีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรม มีขนาด 8 บิต
3. บิตตรวจสอบพาริตี้ (*parity bit*) มีขนาด 1 บิตหรือไม่มี
4. บิตปิดท้ายหรือบิตหยุด (stop bit) มีขนาด 1 บิต



รูปที่ 2.5 แสดงรูปแบบข้อมูลอนุกรมแบบอะซิงโครนัส

รูปที่ 2.5 แสดงรูปแบบของข้อมูล อนุกรมแบบอะซิงโครนัส เมื่อไม่มีการส่งข้อมูล ขา DATA จะมีสถานะลอจิก "1" *เรียกสถานะนี้ว่า สถานะหยุดรอ (waiting stage) การเริ่มต้นส่งข้อมูลจะเริ่มจากการให้ขา* DATA มีลอจิก "0" ด้วยช่วงระยะเวลา 1 บิต เรียกบิตนี้ว่า บิตเริ่มต้น(start bit) จากนั้นบิตข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุดหรือบิต LSB ก่อน ข้อมูลที่ต้องการส่งมีจำนวน 8 บิต ตามด้วยบิตพาริตี้(parity bit) ซึ่งใช้ในการตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล บิตสุดท้ายที่จะส่งคือบิตปิดท้ายหรือบิตหยุด(stop bit) โดยจะเป็นการทำให้ขา DATA มีสถานะลอจิก "1"อีกครั้งด้วยระยะเวลาอย่างน้อย 1 บิต,1.5บิต หรือ 2 บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัตราความเร็วในการรับและส่งข้อมูลของการรับส่งข้อมูลแบบอะซิงโครนัสหรืออัตราบอดหรือบอดเรตที่ใช้สำหรับพอร์ตอนุกรม RS-232 มีด้วยกันหลายค่า ได้แก่ 110, 150, 300, 600, 1200, 2400 ,4800 ,9600 และ 19200 บิตต่อวินาที โดยมีค่าเพิ่มมากขึ้นตามเทคโนโลยีของคอมพิวเตอร์ เนื่องจากอัตราบอดคือค่าของจำนวนบิตที่สามารถส่งได้ใน 1 วินาที สมมติว่า ข้อมูลอนุกรมมีขนาด 8 บิต ไม่มีการตรวจสอบพาริตี้ มีบิตเริ่มต้น 1 บิต และบิตปิดท้าย 1 บิต ความยาวของข้อมูล 1 ไบต์จะเท่ากับความยาว 10 บิต ถ้าใช้บอดเรตในการส่งข้อมูลเท่ากับ 9,600 บิตต่อวินาที ก็สามารถรับส่งข้อมูลได้ด้วยความเร็ว 960 ไบต์ต่อวินาที

การตรวจสอบพาริตี้สามารถกำหนดให้เป็นแบบคี่(odd),แบบคู่(even) หรือไม่มีการตรวจสอบพาริตี้ก็ได้ การตรวจสอบแบบพาริตี้คี่(หรือคู่)คือ การนับจำนวนลอจิก “1” ของข้อมูลขนาด 1 ไบต์และของบิตพาริตี้ว่ามีจำนวนรวมเป็นเลขคี่(หรือคู่)หรือไม่ ถ้าใช่แสดงว่าข้อมูลที่ส่งมาถูกต้อง ยกตัวอย่าง ข้อมูลที่จะทำการส่งมีขนาด 8 บิต มีค่าเท่ากับ 99H หรือ 10011001B จะเห็นได้ว่าข้อมูลในไบต์นี้มีจำนวนลอจิก”1”จำนวน 4 ตัวซึ่งเป็นเลขคู่ ดังนั้นถ้ากำหนดการตรวจสอบค่าพาริตี้เป็นแบบคู่ ค่าของบิตพาริตี้จะต้องมีลอจิกเป็น”0” แต่ถ้ากำหนดการตรวจสอบพาริตี้เป็นแบบคี่ ค่าของบิตพาริตี้จะต้องเป็น”1” เพื่อให้จำนวนลอจิก “1” ข้อมูล 1 ไบต์รวมทั้งในบิตพาริตี้มีจำนวนรวมเป็นคี่

บิตพาริตี้จะถูกสร้างขึ้นจากภาคส่งข้อมูลของ UART(*Universal Asynchronous Receiver Transmitter:* เป็นอุปกรณ์ที่ใช้ในการรับและส่งข้อมูลอนุกรม) ซึ่งทางภาครับจะต้องกำหนดคุณสมบัติการตรวจสอบพาริตี้ที่ตรงกันเอาไว้ว่าตรวจสอบพาริตี้คี่หรือพาริตี้คู่ จากนั้นภาครับของ UART จะทำการตรวจสอบค่าพาริตี้ที่เกิดขึ้นว่าเป็นคู่หรือเป็นคี่ โดยการนับจำนวนลอจิก 1 ทั้งหมดรวมทั้งบิตพาริตี้ด้วย ถ้ากำหนดพาริตี้ไว้เป็นคู่แต่อ่านค่าตัวเลขในการนับออกมาได้ตัวเลขเป็นคี่ ทางภาครับจะแสดงข้อผิดพลาดออกมาให้ผู้ใช้ทราบ กระบวนการดังกล่าวเป็นวิธีการตรวจสอบความผิดพลาดที่เกิดขึ้นในการรับส่งข้อมูลที่ง่ายที่สุด แต่มันสามารถตรวจสอบได้เมื่อมีบิตข้อมูลที่ทำการรับส่งผิดพลาดเพียงบิตเดียวเท่านั้น ถ้าข้อมูลที่ทำการส่งมีบิตที่มีการผิดพลาดมากกว่า 1 บิต การตรวจสอบด้วยวิธีนี้จะไม่ได้ผล สำหรับการตั้งพาริตี้บิตเป็น NONE นั้นทั้งภาครับและภาคส่ง จะไม่มีการตรวจสอบพาริตี้

2.1.5.2 รีจิสเตอร์ที่เกี่ยวข้องกับการทำงานของพอร์ตอนุกรมใน MCS-51

ในการทำงานของวงจรพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 มีรีจิสเตอร์ที่ต้องเกี่ยวข้องอยู่ 2 ตัว มีรายละเอียดดังต่อไปนี้

1. รีจิสเตอร์บัฟเฟอร์ของพอร์ตอนุกรมหรือ SBUF (Serial Data Buffer Register)

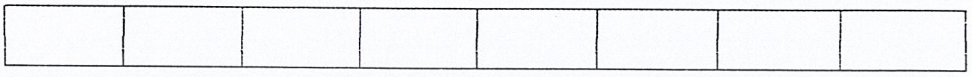
มีแอดเดรสอยู่ที่ 99H ในพื้นที่ของรีจิสเตอร์ฟังก์ชั่นพิเศษหรือ SFR มีขนาด 8 บิต แบ่งเป็น 2 ส่วนคือ รีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล (transmit buffer register) และรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูล (receive buffer register) เมื่อมีการเขียนข้อมูลมายังรีจิสเตอร์ SBUF ข้อมูลนั้นจะถูกส่งต่อไปยังบัฟเฟอร์สำหรับส่งข้อมูล เพื่อส่งออกจากไมโครคอนโทรลเลอร์ผ่านทางขาTxD หรือ P3.1 ในกรณีที่มีการอ่านข้อมูลจากรีจิสเตอร์ SBUF ข้อมูลจะถูกส่งผ่านไปยังรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูลเพื่อส่งต่อไปยังไมโครคอนโทรลเลอร์ต่อไป สำหรับการรับข้อมูลอนุกรมจากภายนอกนั้นจะผ่านมาทางขา RxD หรือ P3.0 ของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. รีจิสเตอร์ควบคุมการทำงานของพอร์ตอนุกรมหรือ SCON

(Serial Port Control Register)

เป็นรีจิสเตอร์ขนาด 8 บิต มีแอดเดรสอยู่ที่ 98H ในพื้นที่ของรีจิสเตอร์ SFR สามารถเข้าถึงได้ในระดับบิต มีรายละเอียดการทำงานดังนี้



รูปที่ 2.6 แสดงรีจิสเตอร์ควบคุมการทำงานของพอร์ตอนุกรม

SMO-SM1 (Serial port mode bit 0-1) : ใช้ในการเลือกโหมดการทำงานของพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์

SM2 : ใช้ในการอีนาเบิลการสื่อสารในแบบมัลติโปรเซสเซอร์ (multiprocessor) ในการทำงานของโหมด 2 และ 3 ของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 ในโหมด 2และ3 ถ้าบิตนี้เป็น"1" บิต RI จะไม่แอกตีฟถ้าบิตที่ 9 ที่รับเข้ามาเป็น "0" (ข้อมูลบิตที่ 9 เก็บไว้ที่บิต RB8) ในการทำงานโหมด 1 ถ้าบิตนี้ถูกเซตบิต RI จะไม่แอกตีฟถ้ายังไม่รับบิตหยุด ส่วนในโหมด 0 บิตนี้ไม่มีการใช้งาน

REN(Enable serial reception): ใช้ในการอีนาเบิลการรับข้อมูลพอร์ตอนุกรม ทำการเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์ ถ้าต้องการรับข้อมูลต้องเซตบิตนี้ให้เป็น "1"

TB8: ใช้สำหรับเก็บข้อมูลที่บิต 9 ที่ต้องการส่งออกไปขณะทำงานในโหมด 2และ3 ของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์MCS-51 เซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

RB8: ใช้สำหรับข้อมูลบิตที่ 9 ที่เข้ามาขณะทำงานในโหมด 2และ3 ของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์MCS-51 แต่ถ้าหากพอร์ตอนุกรมทำงานอยู่ในโหมด 1 และบิต SM2 เป็น"0" ข้อมูลที่บิต RB8 คือข้อมูลของบิตหยุด (stop bit) สำหรับการทำงานในโหมด 0 บิตนี้จะไม่ใช้งาน บิต RB8นี้สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

TI(Transmit Interrupt flag) : ใช้ในการแสดงการเกิดอินเตอร์รัปต์เมื่อมีการส่งข้อมูลออกจากพอร์ตอนุกรมของไมโครคอนโทรลเลอร์MCS-51 สามารถเซตได้ด้วยกระบวนการทางฮาร์ดแวร์ เมื่อทำการส่งข้อมูลบิตที่ 8 เรียบร้อยแล้วในการทำงานโหมด 0 ส่วนการทำงานโหมดอื่น บิตนี้จะเซตเมื่อมีการเริ่มต้นส่งบิตหยุดออกไป การเคลียร์บิตนี้ต้องใช้กระบวนการทางซอฟต์แวร์เท่านั้น

RI(Receive Interrupt flag) : ใช้ในการแสดงการเกิดอินเตอร์รัปต์เมื่อมีการรับข้อมูลเข้าจากพอร์ตอนุกรมของไมโครคอนโทรลเลอร์MCS-51 สามารถเซตได้ด้วยกระบวนการทางฮาร์ดแวร์ เมื่อทำการรับข้อมูลบิตที่ 8 เรียบร้อยแล้วในการทำงานโหมด 0 ส่วนการทำงานโหมดอื่น บิตนี้จะเซตเมื่อมีการรับบิตหยุดของข้อมูลอนุกรมไปได้ครึ่งทางแล้ว ยกเว้นกรณีที่บิต SM2 มีการเซต บิตนี้จะเซตได้ก็ต่อเมื่อการรับบิตหยุดหรือบิตที่ 9 เกิดขึ้นอย่างสมบูรณ์แล้ว การเคลียร์บิตนี้ต้องใช้กระบวนการทางซอฟต์แวร์เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5.3 โหมดการทำงานของพอร์ตอนุกรมใน MCS-51

พอร์ตอนุกรมในไมโครคอนโทรลเลอร์MCS-51 แบบแฟลชสามารถเลือกโหมดการทำงานได้ถึง 4 โหมดคือ

1. โหมด 0 เป็นการกำหนดให้พอร์ตอนุกรมทำงานในลักษณะชิฟต์รีจิสเตอร์
2. โหมด 1 เป็นการกำหนดให้เป็น UART ขนาด 8 บิต สามารถเลือกอัตราบอดได้
3. โหมด 2 เป็นการกำหนดให้เป็น UART ขนาด 9 บิต โดยมีอัตราบอดคงที่
4. โหมด 3 เป็นการกำหนดให้เป็น UART ขนาด 9 บิต สามารถเลือกอัตราบอดได้

การเลือกโหมดการทำงานของวงจรพอร์ตอนุกรมในไมโครคอนโทรลเลอร์MCS-51 กระทำได้โดยการกำหนดข้อมูลให้แก่บิต SMO และ SM1 ในรีจิสเตอร์ SCON

*2.1.5.4 อัตราบอดของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์*MCS-51

1. โหมด 0

อัตราบอดของโหมดนี้มีค่าคงที่ โดยสามารถคำนวณได้จากสูตร

อัตราบอดใน โหมด 0 = ความถี่ของสัญญาณนาฬิกา/12 มีหน่วยเป็นบิตต่อวินาที

2. โหมด 1 และ 3

ทั้งสองโหมดนี้สามารถเลือกแหล่งกำเนิดอัตราบอดได้ 2 แหล่งคือ จากอัตราโอเวอร์โฟลวของไทเมอร์ 1 และ 2 สำหรับอัตราบอดเมื่อใช้การโอเวอร์โฟลวของไทเมอร์ 1 จะต้องใช้ค่าของบิต SMOD ในรีจิสเตอร์ PCON มาพิจารณาประกอบด้วย โดยสามารถหาค่าอัตราบอดได้จาก

อัตราบอด = $(2^{\text{ค่าของบิต SMOD}}/32)$ X อัตราโอเวอร์โฟลวของไทเมอร์ 1

อัตราบอด (บิตต่อวินาที : bps)	ความถี่ สัญญาณนาฬิกา	SMOD	ไทเมอร์ 1		
			C/T	โหมด	ค่ารีโหลด
โหมด 0 : สูงสุด 1 MHz	12 MHz	x	x	x	x
โหมด 2 : สูงสุด 375k	12 MHz	1	x	x	x
โหมด 1,3 : 62.5 K	12 MHz	1	0	2	FFH
19.2K (19,200)	11.0592 MHz	1	0	2	FDH
9.6K (9,600)	11.0592 MHz	0	0	2	FDH
4.8K (4,800)	11.0592 MHz	0	0	2	FAH
2.4K(2,400)	11.0592 MHz	0	0	2	F4H
1.2K (1,200)	11.0592 MHz	0	0	2	E8H
137.5	11.0592 MHz	0	0	2	1DH
110	6 MHz	0	0	2	72H
110	12 MHz	0	0	1	FEEBH

ตารางที่ 2.2 แสดงการเลือกอัตราบอดของวงจรพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าหากในไทเมอร์1ไม่ได้อีนาเบิลการอินเตอร์รัปต์ไว้ สามารถคำนวณหาค่าอัตราบอดได้จาก

อัตราบอด = ($2^{\text{ค่าในรีจิสเตอร์ SMOD}}$/32) X (ความถี่สัญญาณนาฬิกา/{12x[256-(TH1)]})

ตารางที่2-2 แสดงการกำหนดอัตราบอดโดยใช้ไทเมอร์ 1

กรณีที่ใช้ไทเมอร์ 2 ในการกำหนดอัตราบอด โดยกำหนดให้ไทเมอร์ 2 ทำงานในโหมดกำเนิดอัตราบอด (baud rate generator) สามารถคำนวณหาค่าอัตราบอดได้จาก

อัตราบอด = อัตราโอเวอร์โฟลวของไทเมอร์ 2/16 มีหน่วยเป็นบิตต่อวินาที

ถ้าหากกำหนดให้ไทเมอร์ 2 ทำงานในโหมดไทเมอร์หรือเคาน์เตอร์ตามปกติ สามารถคำนวณหาค่าอัตราบอดได้จาก

อัตราบอด = ความถี่สัญญาณนาฬิกา/(32 X (65536-(RCAP2H,RCAP2L)))

โดยที่(RCAP2H,RCAP2L) เป็นค่าของรีจิสเตอร์ RCAP2H และ RCAP2L มีขนาด 16 บิตไม่คิดเครื่องหมาย

3. โหมด 2

ในโหมดนี้การกำหนดอัตราบอดจะขึ้นอยู่กับค่าของบิต SMOD ในรีจิสเตอร์ PCON ถ้า SMOD เป็น"0" อัตราบอดจะเท่ากับ 1/64 ของความถี่สัญญาณนาฬิกา ในกรณีที่ SMOD เป็น "1" อัตราบอดจะเท่ากับ 1/32 ของความถี่สัญญาณนาฬิกา สามารถแสดงเป็นสูตรคำนวณทางคณิตศาสตร์ได้ดังนี้

อัตราบอด = ($2^{\text{ค่าของบิต SMOD}}$/64) x ความถี่สัญญาณนาฬิกา

2.1.5.5 การกำหนดค่าของไทเมอร์เพื่อเลือกอัตราบอด

ในการใช้งานพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 สิ่งที่ต้องให้ความสนใจมากที่สุดประการหนึ่งคือ อัตราการถ่ายทอดข้อมูล หรือ อัตราบอด ซึ่งการกำหนดอัตราบอดนั้นจะขึ้นอยู่กับค่าความถี่ของสัญญาณนาฬิกาเป็นหลัก สำหรับโหมดการทำงานของพอร์ตอนุกรมที่สามารถเลือกอัตราบอดได้อย่างอิสระคือโหมด 1 และ 3 โดยกำหนดได้จากอัตราการเกิดโอเวอร์โฟลวของไทเมอร์ 1 หากไทเมอร์ 1 มีการเกิดโอเวอร์โฟลวในอัตราที่สูงมากอัตราบอดก็จะมีค่าสูงมากขึ้นตาม นั่นหมายความว่า อัตราในการถ่ายทอดข้อมูลจะสูงมาก สามารถถ่ายทอดข้อมูลได้อย่างรวดเร็ว ดังนั้นการกำหนดค่าและโหมดการทำงานของไทเมอร์ 1 จึงเป็นสิ่งที่มีผลต่อการเปลี่ยนแปลงของอัตราบอดด้วย

ในการใช้ไทเมอร์1 เพื่อกำหนดอัตราบอดในโหมด1และ3 ของพอร์ตอนุกรมจะต้องกำหนดให้ไทเมอร์ ทำงานในโหมด 2 หรือโหมด 8 บิตแบบตั้งค่าการนับอัตโนมัติ ในกรณีการกำหนดค่ารีโหลดให้แก่รีจิสเตอร์ TH1 จะเป็นตัวแปรหลักที่ใช้ในการกำหนดอัตราบอดให้แก่พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51

เริ่มต้นด้วยการเคลียร์บิต SMOD ซึ่งเป็นบิต 7 ของรีจิสเตอร์ PCON ให้เป็น "0" ค่าของการรีโหลดให้แก่ TH1 สามารถคำนวณได้จาก

TH1 = 256-((ค่าความถี่ของคริสตอล/384)/อัตราบอด)

แต่ถ้าบิต SMOD เกิดจากการเซต จะเป็นการอีนาเบิลการทวีคูณของอัตราบอด ดังนั้นการกำหนดค่าให้แก่ TH1 จึงต้องคำนวณจาก

TH1 = 256-((ค่าความถี่ของคริสตอล/192)/อัตราบอด)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยกตัวอย่าง ถ้าหากในไมโครคอนโทรลเลอร์ AT89C51 ใช้คริสตอล 11.0592 MHz ต้องการกำหนดอัตราบอดของพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ไว้ที่ 19,200 บิตต่อวินาที ในกรณีที่ไม่อีนาเบิลการทวีคูณของอัตราบอด ค่ารีโหลดของไมโครคอนโทรลเลอร์จะเท่ากับ

TH1 = 256-((ค่าความถี่ของคริสตอล/384)/อัตราบอด

= 256-((11059200/384)/19200)

= 256-(28800/19200)

= 256-1.5 = 254.5

เนื่องจากผลลัพธ์ที่ได้เป็นค่าที่ไม่ใช่จำนวนเต็ม ถ้าหากกำหนดค่าของ TH1 เป็น 254 เมื่อทำการแทนค่าเพื่อคำนวณหาค่าอัตราค่าบอด จะได้อัตราบอดเท่ากับ 14,400 บิตต่อวินาที และถ้าหากกำหนดค่าของ TH1 เป็น 255 อัตราบอดจะมีค่าเท่ากับ 28,800 บิตต่อวินาที ดังนั้นจะเห็นได้ว่าค่าของ TH1 ที่ไม่เป็นจำนวนเต็มจะไม่สามารถทำให้เกิดอัตราบอดตามต้องการได้

ทางแก้ไขคือ ให้ทำการอีนาเบิลการทวีคูณอัตราบอด โดยการเซตบิต SMOD ในรีจิสเตอร์ PCON ให้เป็น “1” จากนั้นทำการแทนค่าลงในสมการหาค่า TH1 เมื่อมีการเซตบิต SMOD ได้ผลดังนี้

TH1 = 256-((ค่าความถี่ของคริสตอล/192)/อัตราบอด)

= 256-((11059200/192)/19200)

= 256-(57600/19200)

= 256-3 = 253

นำค่าของ TH1 ที่ได้ทำการแทนค่าคำนวณหาค่าอัตราบอดจะได้เท่ากับ 19,200 บิตต่อวินาที

สามารถสรุปขั้นตอนในการเลือกอัตราบอดโดยการกำหนดค่าของไทเมอร์ 1 ได้ดังนี้

1. กำหนดให้พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 ทำงานในโหมด 1 หรือ 3
2. กำหนดให้ไทเมอร์ 1 ทำงานในโหมด 2 หรือโหมด 8 บิตตั้งค่าอัตโนมัติ
3. กำหนดข้อมูลให้แก่ TH1 เท่ากับ 253 เพื่อให้สามารถกำเนิดอัตราบอดได้ 19,200 บิตต่อวินาที ตามที่ต้องการ
4. ทำการเซตบิต SMOD ซึ่งเป็นบิต 7 ของรีจิสเตอร์ PCON เพื่ออีนาเบิลการทวีคูณอัตราบอด

2.1.5.6 การเขียนหรือส่งข้อมูลออกจากพอร์ตอนุกรม

ข้อมูลที่ต้องการส่งออกทุกค่าต้องนำไปเก็บไว้ในรีจิสเตอร์บัฟเฟอร์ของพอร์ตอนุกรม ซึ่งก็คือ รีจิสเตอร์ SBUF ดังตัวอย่าง

```
MOV SBUF , #' A'
```

จากคำสั่งข้างต้นเป็นการส่งข้อมูลของตัวอักษร A ออกไปยังพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ อย่างไรก็ตามก่อนที่จะส่งข้อมูลทุกครั้ง ต้องแน่ใจว่าบิต TI เคลียร์ หรือ มีค่าเป็น “0” และเมื่อทำการส่งข้อมูลเรียบร้อยแล้ว ก็จะเกิดการเซตบิต TI เพื่อแจ้งให้ทราบ ดังตัวอย่างโปรแกรมต่อไปนี้

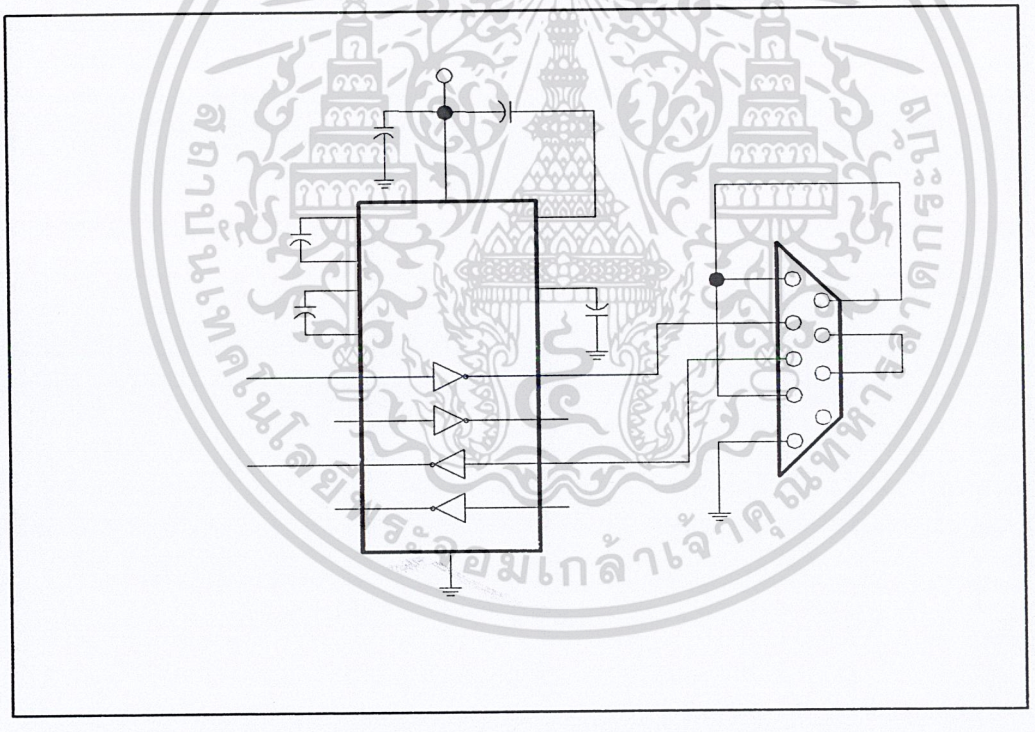
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CLR TI                  ; เคลียร์บิต TI เพื่อเตรียมการส่งข้อมูลออก
MOV SBUF, #'A'          ; ส่งข้อมูลของตัวอักษร A ไปยังพอร์ตอนุกรม
JNB TI , $              ; รอการเซตของบิต TI เพื่อแจ้งการส่งข้อมูลที่เสร็จสมบูรณ์
```

2.1.5.7 การอ่านหรือรับข้อมูลจากพอร์ตอนุกรม

การรับข้อมูลจากพอร์ตอนุกรมสามารถกระทำได้ง่ายมาก เพียงแค่ทำการตรวจสอบว่าบิต RI เกิดการเซตขึ้นหรือไม่ ถ้าพบว่ามีการเซตเกิดขึ้นแล้ว ให้ทำการอ่านค่าจากรีจิสเตอร์ SBUF โดยต้องทำการโอนย้ายข้อมูลผ่านทางแอกคิวมูเลเตอร์หรือรีจิสเตอร์ A ดังตัวอย่าง

```
CLR RI          ; เคลียร์บิต RI เพื่อเตรียมการรับข้อมูล
JNB RI , $      ; รอคอยการเซตของบิต RI อันเป็นการแจ้งให้ทราบว่า การรับข้อมูล
                ; เสร็จสมบูรณ์และมีข้อมูลที่เกิดขึ้นที่รีจิสเตอร์ SBUF
MOV A , SBUF    ; อ่านค่าจากรีจิสเตอร์ โดยการโอนย้ายข้อมูลผ่านทางรีจิสเตอร์ A
CLR RI          ; หลังจากทำการอ่านข้อมูลเรียบร้อยแล้ว ต้องทำการเคลียร์บิต RI เสมอ
```



รูปที่ 2.7 แสดงวงจรเชื่อมต่อ MAX232 หรือ ICL232 เข้ากับพอร์ตอนุกรมของคอมพิวเตอร์และไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5.8 การเชื่อมต่อกับพอร์ตอนุกรมของคอมพิวเตอร์

การใช้งานวงจรพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 มักนิยมใช้ในการติดต่อเพื่อแลกเปลี่ยนข้อมูลกับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรมในมาตรฐาน RSS-232 เป็นส่วนใหญ่ แต่เนื่องจากระดับสัญญาณของพอร์ตอนุกรม RS-232 มีระดับตั้งแต่ 3 ถึง 12 V ในขณะที่ระดับสัญญาณของไมโครคอนโทรลเลอร์ MCS-51 อยู่ในระดับทีทีแอล ดังนั้นจึงไม่สามารถเชื่อมต่อพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 เข้ากับพอร์ตอนุกรมของคอมพิวเตอร์ได้โดยตรง จึงต้องอาศัยการเชื่อมต่อผ่านไอซีพิเศษที่ทำหน้าที่แปลงระดับสัญญาณ

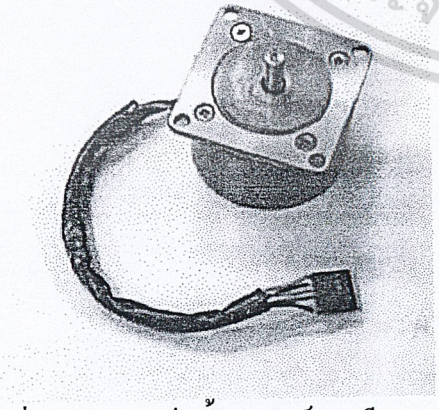
ไอซีที่ทำหน้าที่ในการแปลงระดับสัญญาณนี้จะต้องทำการแปลงข้อมูลส่งของไมโครคอนโทรลเลอร์ MCS-51 จากระดับทีทีแอลไปเป็นระดับของ RS-232 และทำการแปลงข้อมูลที่รับจากคอมพิวเตอร์จากระดับของ RS-232 เป็นระดับทีทีแอลเพื่อให้สามารถถ่ายทอดไปยังไมโครคอนโทรลเลอร์ MCS-51 ได้อย่างสมบูรณ์ ไอซีดังกล่าวมีด้วยกันหลายเบอร์จากผู้ผลิต อาทิ MAX232 จาก MAXIM หรือ ICL232 จาก HARRIS เป็นต้น ในรูปที่ 2.7 แสดงส่วนวงจรของการต่อกับพอร์ตอนุกรมของคอมพิวเตอร์และไมโครคอนโทรลเลอร์

***2.2 สเต็ปปิ้งมอเตอร์* (Stepping Mortor)**

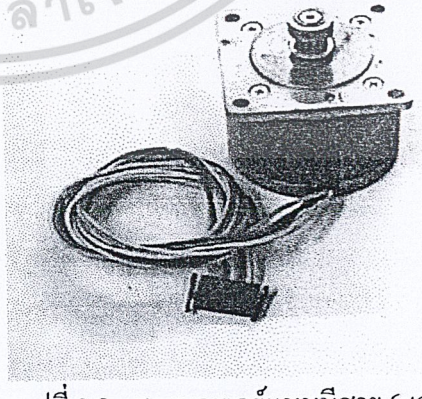
สเต็ปปิ้งมอเตอร์ เป็นมอเตอร์ที่ขับเครื่องด้วยพัลส์ ลักษณะการขับเคลื่อน จะหมุนรอบแกนได้ 360 องศา มีลักษณะไม่ต่อเนื่อง แต่มีลักษณะเป็นสเต็ป โดยแต่ละสเต็ปจะขับเคลื่อนได้ 1,1.5,1.8 หรือ 2 องศา แล้วแต่ละโครงสร้างของมอเตอร์และลักษณะที่ นำมอเตอร์ไปใช้ ตัวอย่างงานที่ต้องการตำแหน่งแม่นยำเช่น ระบบขับเคลื่อนหัวแม่พิมพ์ในเครื่องพิมพ์ (PRINTER)ระบบขับเคลื่อนหัวอ่านในเครื่องอ่านบันทึกเหล็ก ระบบขับเคลื่อนตำแหน่งของปากกาใน X-Y PLOTER เป็นต้น

2.2.1 ลักษณะและโครงสร้างของสเต็ปปิ้งมอเตอร์

สเต็ปปิ้งมอเตอร์มีลักษณะดังรูป

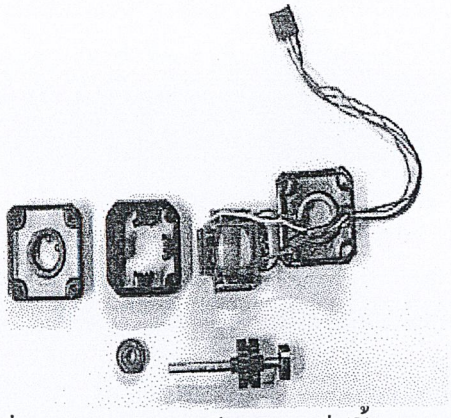


รูปที่ 2.8 แสดงสเต็ปปิ้งมอเตอร์แบบมีสาย 5 เส้น

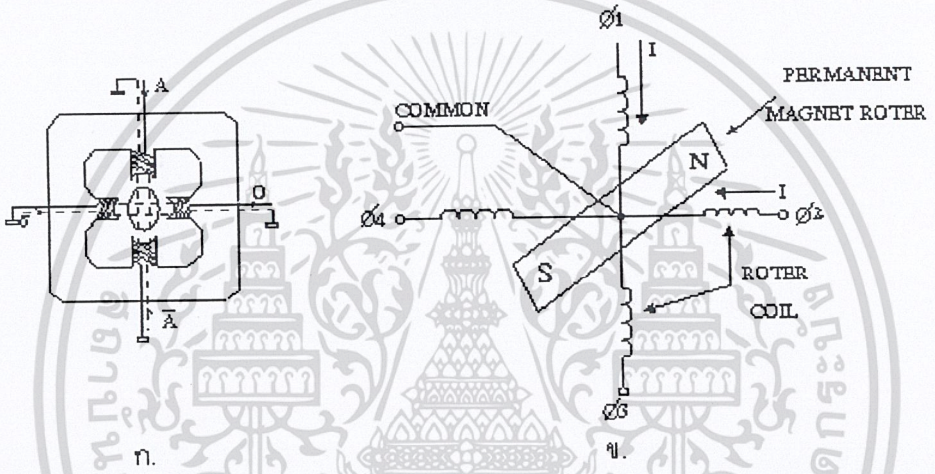


รูปที่ 2.9 แสดงมอเตอร์แบบมีสาย 6 เส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 แสดงโครงสร้างของสเต็ปปิ้งมอเตอร์



รูป 2.11 แสดง (ก) โครงสร้าง (ข) วงจรเทียบเท่า (equivalent circuit) ของมอเตอร์ ชนิด 4 ขด

2.2.2 ชนิดของสเต็ปปิ้งมอเตอร์

2.2.2.1 แบบแม่เหล็กถาวร **(PERMANENT MAGNET_PM)** สเต็ปปิ้งมอเตอร์แบบ PM จะมีสเตเตอร์ (STATOR) ที่พันขดลวดไว้หลายๆ โพล โดยมีโรเตอร์ (ROTOR) เป็นรูปทรงกระบอกฟันเลื่อย และโรเตอร์ทำด้วยแม่เหล็กถาวร เพื่อป้อนไฟกระแสตรงให้กับขดสเตเตอร์ ทำให้เกิดแรงแม่เหล็กไฟฟ้าผลักต่อโรเตอร์และ ทำให้มอเตอร์หมุน มอเตอร์แบบ PM จะเกิดแรงฉุดยึดให้โรเตอร์หยุดอยู่กับที่ แม้จะไม่ได้ป้อนไฟเข้าขดลวด

2.2.2.2 แบบแปรค่ารีลักแตนซ์ **(VARIABLE RELUCTANCE- VR)** สเต็ปปิ้งมอเตอร์แบบ VR จะมีการหมุนโรเตอร์ได้อย่างอิสระ แม้จะไม่ได้จ่ายไฟให้โรเตอร์ ทำจากสารเฟอร์โรแมกเนติกกำลังอ่อน มีลักษณะเป็นฟันเลื่อย รูปทรงกระบอก โดยจะมีความสัมพันธ์โดยตรงกับจำนวนโพลในสเตเตอร์ แรงบิดที่เกิดขึ้นจะหมุนโรเตอร์ไปในเส้นทางของอำนาจแม่เหล็กที่มีค่ารีลักแตนซ์ต่ำที่สุด ตำแหน่งที่จะเกิดแน่นอนและมีเสถียรภาพแต่จะเกิดขึ้นได้หลายๆ จุดดังนั้นเมื่อป้อนไฟเข้าขดลวดต่างๆ ในมอเตอร์แตกต่างขดกันไป ก็ทำให้มอเตอร์หมุนไปตำแหน่งต่างๆ กัน โรเตอร์ของ VR จะมีความเฉื่อยของโรเตอร์น้อยจึงมีความเร็วรอบสูงกว่ามอเตอร์แบบ PM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.2.2.3 แบบผสม **(HYBRID-H)**สเต็ปปิ้งมอเตอร์แบบ H จะเป็นลูกผสมของ VR กับ PM โดยจะมีสเตเตอร์คล้ายกับที่ใช้ใน VR โรเตอร์มีหมวกหุ้ม ปลายซึ่งมีลักษณะของสารแม่เหล็กที่มีกำลังสูง โดยการควบคุมขนาดรูปร่างของหมวกแม่เหล็กอย่างดีจะทำให้ได้มุม การหมุนและจำนวนครั้งในการหมุนน้อยและแม่นยำ ข้อดีก็คือ ให้แรงบิดสูงและมีขนาดกระทัดรัด และให้แรงฉุดยึดโรเตอร์นิ่งกับที่ตอนไม่จ่ายไฟ

2.2.3 การกระตุ้นและควบคุมการหมุนของสเต็ปปิ้งมอเตอร์

การกระตุ้นและควบคุมการหมุนของมอเตอร์ให้เคลื่อนที่ไปแต่ละสเต็ปทำได้โดยจ่ายกำลังไฟฟ้าไปยังขดลวดแต่ละขดบนสเตเตอร์ ซึ่งต้องป้อนเป็นแบบซีเควนเชียล(Sequential)ในรูปแบบที่ถูกต้องด้วย สามารถแบ่งได้เป็น 3 รูปแบบคือ

2.2.3.1 แบบหนึ่งเฟสหรือแบบฟูลสเต็ป(**Full step**) เป็นการกระตุ้นโดยทำการกระตุ้นขดลวดทีละขดในเวลาหนึ่งไล่เรียงถัดกันไป ทำให้ในการกระตุ้นรูปแบบนี้จึงมีขดลวดเพียงขดลวดเดียวในเวลาหนึ่งที่ถูกกระตุ้นเท่านั้น วงจรแบบเวฟจึงมีราคาถูกและง่าย ขั้นตอนการทำงานต่างๆ แสดงดังตารางที่ 2.3

สเต็ปที่	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	ทำงาน	-	-	-
2	-	ทำงาน	-	-
3	-	-	ทำงาน	-
4	-	-	-	ทำงาน

ตารางที่ 2.3 แสดงการควบคุมสเต็ปปิ้งมอเตอร์แบบหนึ่งเฟสหรือแบบฟูลสเต็ป(Full step)

2.2.3.2 แบบ 2 เฟส เป็นการกระตุ้นคล้ายแบบหนึ่งเฟส แต่จะทำการกระตุ้นโดยจ่ายกำลังไฟฟ้าไปที่ขดลวด 2 ขด ที่อยู่ใกล้กันในเวลาเดียวกัน และเรียงถัดไปเช่นเดียวกับแบบเวฟ การกระตุ้นสเต็ปปิ้งมอเตอร์แบบนี้สามารถเพิ่มแรงบิดได้มากกว่าแบบเวฟ โรเตอร์จะเคลื่อนที่ด้วยแรงดึงอย่างเต็มแรงจาก 2 ขดลวดที่ถูกกระตุ้นพร้อมกัน และต่อไปด้วยแรงดึงจากอีก 2 ขดลวดถัดไป สำหรับข้อเสียคือต้องใช้กำลังไฟฟ้ามากขึ้น ขั้นตอนการทำงานต่างๆ แสดงดังตารางที่ 2.4

สเต็ปที่	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	ทำงาน	ทำงาน	-	-
2	-	ทำงาน	ทำงาน	-
3	-	-	ทำงาน	ทำงาน
4	ทำงาน	-	-	ทำงาน

ตารางที่ 2.4 แสดงการควบคุมสเต็ปปิ้งมอเตอร์แบบ 2 เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

49985

2.2.3.3 แบบครึ่งสเต็ป เป็นรูปแบบการผสมผสานระหว่างการกระตุ้นแบบเวฟและแบบ 2 เฟส เพื่อเพิ่มจำนวนของสเต็ปต่อรอบอีกเท่าตัวหนึ่ง ในระบบนี้จะทำการกระตุ้นเรียงกันไปเป็นลำดับดังนี้ เริ่มจากขดลวดที่ 1, 1 และ 2, 2, 2 และ 3 ,3 ,3 และ 4, 4, 4 และ 1 แล้ววนกลับมายังขดลวดที่ 1 แรงบิดที่ได้จากการกระตุ้นแบบนี้จะเพิ่มมากขึ้นอีก เพราะช่วงสเต็ปมีระยะสั้นลง แต่ละสเต็ปเกิดแรงดึงจากขดลวด 2 ขดที่ถูกกระตุ้นพร้อมกัน ความถูกต้องของตำแหน่งมีเพิ่มมากขึ้น แต่ต้องพึงระวังเรื่อง เมื่อกระตุ้นให้ทำงานในรูปแบบนี้จะต้องหมุนถึง 2 สเต็ป จึงจะได้ระยะเท่ากับ 1 สเต็ปเต็มของการควบคุมใน 2 แบบแรก สำหรับแหล่งจ่ายไฟฟ้าต้องใช้ขนาดเท่ากับแบบ 2 เฟสเป็นอย่างน้อยจึงจะพอ ขั้นตอนการทำงานต่างๆ แสดงดังตารางที่ 2.5

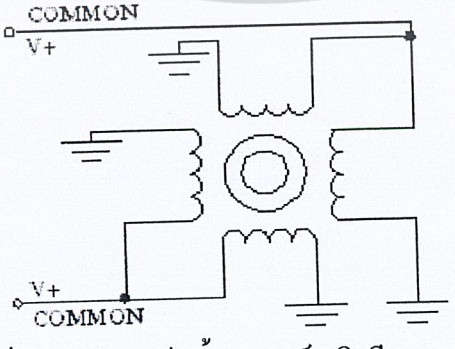
สเต็ปที่	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	ทำงาน	-	-	-
2	ทำงาน	ทำงาน	-	-
3	-	ทำงาน	-	-
4	-	ทำงาน	ทำงาน	-
5	-	-	ทำงาน	-
6	-	-	ทำงาน	ทำงาน
7	-	-	-	ทำงาน
8	ทำงาน	-	-	ทำงาน

ตารางที่ 2.5 แสดงการควบคุมสเต็ปปิ้งมอเตอร์แบบครึ่งสเต็ป

2.2.4 *การตรวจสอบหาสาย* COMMON *และสาย* GROUND *ของ* STEPPING *แบบ* PM *(แบบแกนโรเตอร์เป็นแม่เหล็กถาวร)*

โดยทั่วไป SP MOTOR แบบ PM จะมีอยู่ 2 ชนิด

1) ชนิดที่เป็น COMMON ภายนอก SP MOTOR แบบนี้มีสายอยู่ 6 เส้น ดังรูปที่ 2.12



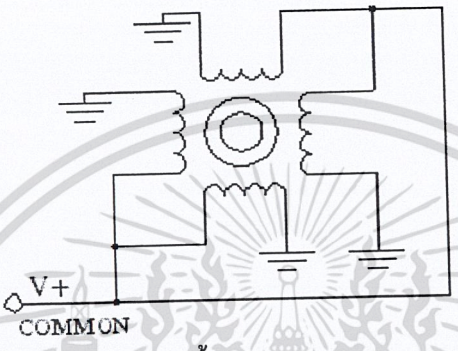
รูปที่ 2.12 แสดงสเต็ปปิ้งมอเตอร์ ชนิดมีสาย 6 เส้น

จากรูปที่ 2.12 สายที่เป็น COMMON มี 2 สาย และสายที่เป็น GROUND มี 4 สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สาย COMMON 1 เส้น จะ DRIVE GROUND 2 เส้น ในการเช็คให้ใช้มิเตอร์วัดหาสายที่เป็น COMMON ก่อนโดยการตั้ง RANGE ของมิเตอร์ที่ R*1 จับที่สายทีละคู่ ถ้าหากวัดสาย COMMON เทียบกับสาย GROUND ได้ถูกต้อง ค่าความต้านทานที่อ่านได้จะน้อย แต่ถ้าวัดผิดสาย คือวัดสาย GROUND เทียบกับ GROUND ค่าความต้านทานที่อ่านได้จะสูงกว่า แต่ถ้าวัดสาย COMMON เทียบกับสาย GROUND ที่ไม่ใช่คู่กันแล้ว เข็มมิเตอร์ก็จะไม่กระดิก ให้ทดลองวัดเปรียบเทียบกันทีละคู่ ก็จะทราบว่าสายใดเป็นสาย COMMON สายใดเป็นสาย GROUND

2) ชนิดที่เป็น COMMON ภายใน SP MOTOR แบบนี้มีสายอยู่ 5 เส้น ดังรูปที่ 2.13



รูปที่ 2.13 แสดงสเต็ปปิ้งมอเตอร์ ชนิดมีสาย 5 เส้น

จากรูปที่ 2.13 สายที่เป็น COMMON มี 1 สาย และสายที่เป็น GROUND มี 4 สาย

ในการวัดให้ทำแบบเดียวกับการวัด SP MOTOR ชนิด COMMON ภายนอก แตกต่างกันเพียงแบบ COMMON ภายใน สาย COMMON 1 เส้น DRIVE สาย GROUND 4 เส้น ดังนั้นหากสายเส้นใดเมื่อวัดเทียบกับสายเส้นอื่น แล้วมีค่าความต้านทานน้อยที่สุดสายเส้นนั้นเป็นสาย COMMON และที่เหลืออีก 4 เส้น จะเป็นสาย GROUND

2.2.5 การเรียงเฟสของ STEPPING MOTOR *แบบ* PM

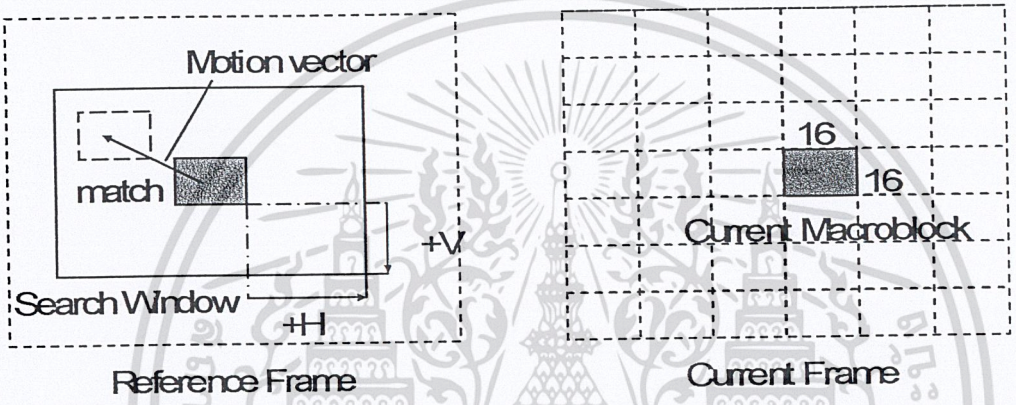
เมื่อเราทราบว่าสายเส้นใดเป็นสาย COMMON แล้ว แต่เรายังไม่ทราบว่าสาย GROUND เส้นใดเป็นเฟสที่1 เฟสที่ 2 เฟสที่ 3 และเฟสที่ 4 ในการเรียงเฟสนั้นให้ใช้มิเตอร์วัดโดยนำ V+ เข้าที่สาย COMMON วัดเทียบกับสาย GROUNDเส้นใดก็ได้ 1 เส้น จะทำให้แกนโรเตอร์เคลื่อนไปข้างหน้า 1 STEP เมื่อเปลี่ยนสาย GROUND เส้นแรกเป็นเส้นที่ 2 หาก MOTOR ไม่เคลื่อนที่ไปข้างหน้าแสดงว่าการเรียงเฟสไม่ถูกต้องก้ให้วัดเทียบกับสาย GROUNDเส้นใหม่ต่อไป หาก MOTOR เคลื่อนที่ไปข้างหน้าตามกัน ก็ให้วัดที่สาย GROUND เส้นต่อไปเรื่อย ๆ จะทำให้ทราบว่าสายเส้นใดเป็นเฟสแรก สายเส้นใดเป็นเฟสที่ 2 เฟสที่ 3 และเฟสที่ 4 การเรียงเฟสของ SP MOTOR แบบ PM ทั้งชนิดที่เป็น COMMON ภายนอกและชนิดที่เป็น COMMON ภายใน ใช้หลักการเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ***การประมวลผลภาพเพื่อทำการตรวจสอบการเคลื่อนที่ของวัตถุ*** มีดังนี้

2.3.1 ***ใช้วิธีเก็บพื้นหลังไว้ก่อน*** คือเมื่อมีวัตถุเคลื่อนที่เข้ามาอยู่ในมุมมองก็ใช้ พื้นหลังที่เก็บไว้ก่อนมาเปรียบเทียบเพื่อหาตำแหน่ง ของวัตถุที่เคลื่อนที่

2.3.2 ***ใช้*** **Block-Matching Algorithm** สำหรับการประมาณการเคลื่อนที่ของวัตถุ จะเห็นดังรูปที่ 2.14 หลักการคือจะหาส่วนที่เข้ากันได้ (Match) ของมาโครบล็อค(Macroblock) ในเฟรมปัจจุบันกับ เฟรมอ้างอิง ซึ่งอาจจะเป็นเฟรมก่อนหน้า โดยจะเข้าไปค้นหาในพื้นที่ ที่เรียกว่า พื้นที่ค้นหา (Scarch Area) หรือวินโดว์ค้นหา (Search Window) เมื่อเจอส่วนที่เข้ากันได้แล้ว ระยะและทิศทางของตำแหน่งส่วนที่เจอกับตำแหน่งของ มาโครบล็อค คือ เวกเตอร์ของการเคลื่อนที่



รูปที่ 2.14 แสดงการใช้มาโครบล็อคจากเฟรมปัจจุบันเข้าไปค้นหาส่วนที่เข้ากันได้ในวินโดว์ค้นหา ภายในเฟรมอ้างอิง จนเจอส่วนที่ เข้ากันได้แล้วก็จะได้เวกเตอร์ของการเคลื่อนที่

2.3.3 ***วิธีที่ประยุกต์ขึ้นมา*** ซึ่งเป็นวิธีที่ใช้ในการตรวจสอบการเคลื่อนที่ของวัตถุในระบบนี้

วิธีนี้อาศัยข้อมูลภาพการเคลื่อนที่ของวัตถุ 3 เฟรมด้วยกัน จะให้เป็น ได้แก่ เฟรมที่ Ft, Ft-1 และ Ft-2 โดยที่ Ft เป็นเฟรมปัจจุบัน, Ft-1 เป็นเฟรมก่อนหน้า Ft และ Ft-2 เป็นเฟรมก่อนหน้าเฟรม Ft-1 ดังตัวอย่างเช่นรูปที่ 2.15, 2.16 และ 2.17 สำหรับ Ft-2, Ft-1 และ Ft ตามลำดับ

ขั้นตอนการตรวจสอบการเคลื่อนที่ของวัตถุ ประกอบด้วย

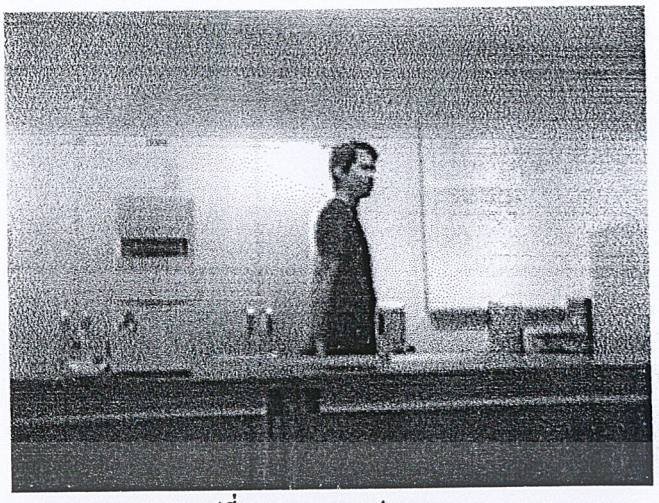
1. สร้างสี่เหลี่ยมครอบบริเวณที่ไม่เหมือนกันของ Ft-2 กับ Ft-1 ดังรูปที่ 2.18

2. สร้างสี่เหลี่ยมครอบบริเวณที่ไม่เหมือนกันของ Ft-1 กับ Ft ดังรูปที่ 2.19

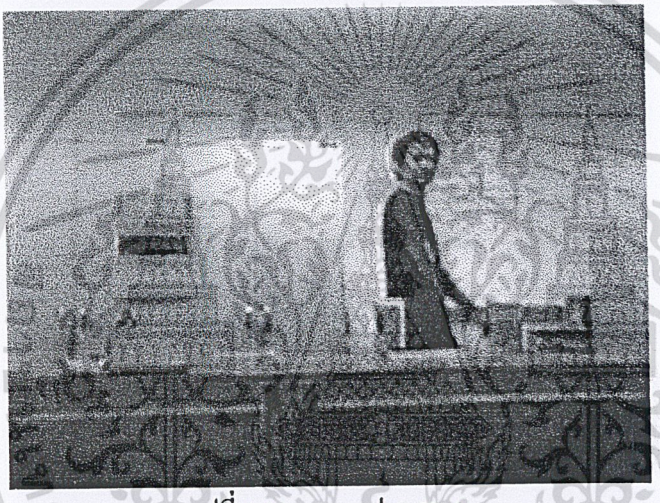
3. นำสี่เหลี่ยมที่ได้ทั้งสองมาตรวจสอบหาบริเวณที่ทับกัน สี่เหลี่ยมที่ได้จากส่วนที่ทับกันนี้คือสี่เหลี่ยมที่ครอบวัตถุที่เคลื่อนที่อยู่นั่นเอง ดังรูปที่ 2.20

4. นำสี่เหลี่ยมที่ได้จากข้อ 3 มาวางในมุมตรงกันข้ามใน สี่เหลี่ยมที่ได้จากข้อ 2 ก็จะได้ตำแหน่งและขนาดของวัตถุโดยประมาณในเฟรมปัจจุบัน(Ft) ดังรูปที่ 2.21

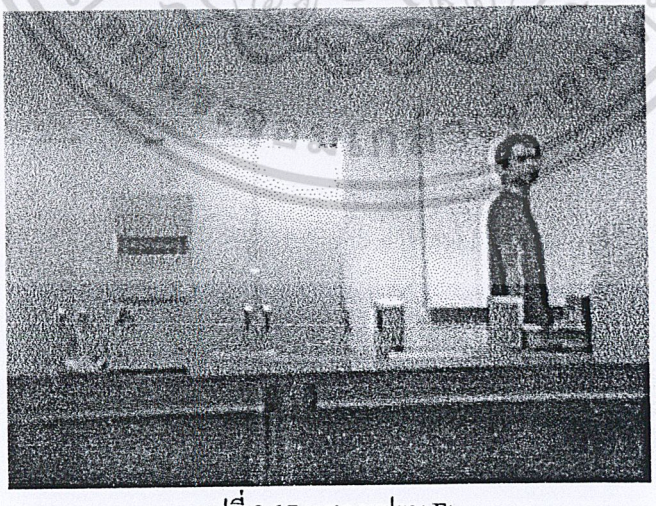
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 แสดงเฟรม Ft-2



รูปที่ 2.16 แสดงเฟรม Ft-1

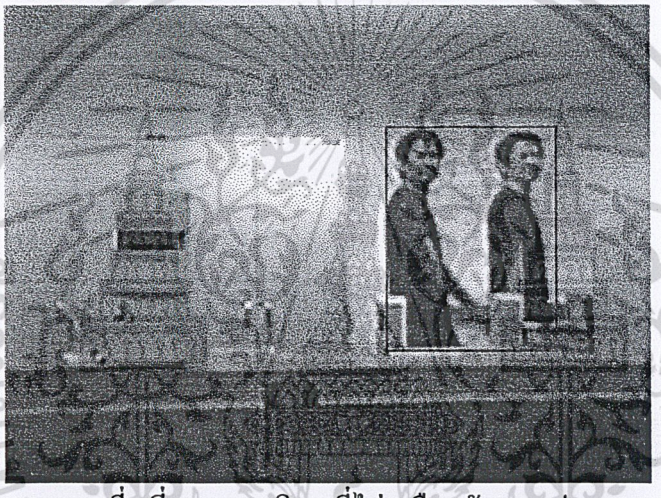


รูปที่ 2.17 แสดงเฟรม Ft

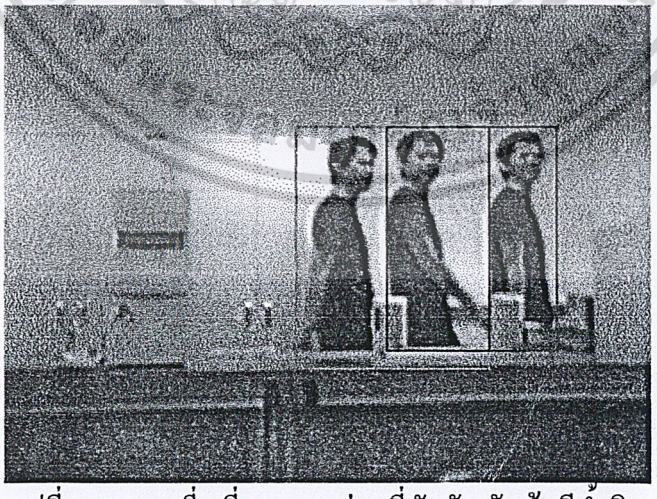
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 แสดงสี่เหลี่ยมครอบบริเวณที่ไม่เหมือนกันของเฟรม Ft-2 กับ Ft-1

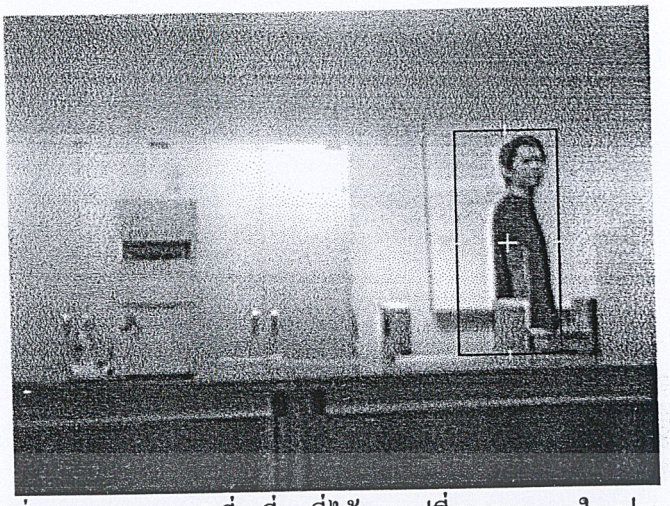


รูปที่ 2.19 แสดงสี่เหลี่ยมครอบบริเวณที่ไม่เหมือนกันของเฟรม Ft-1 กับ Ft



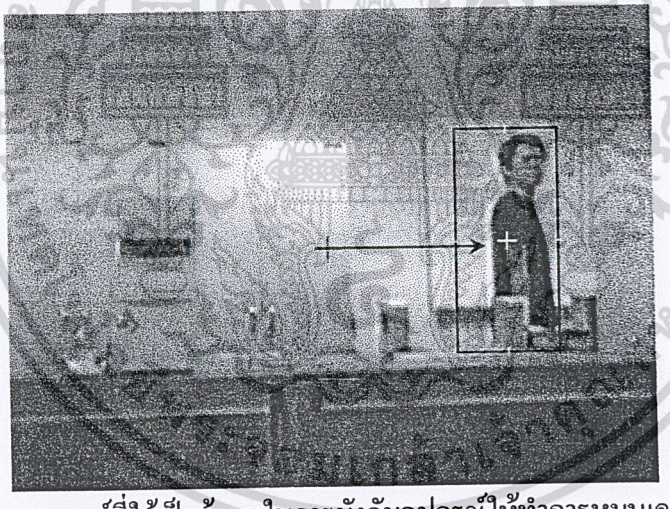
รูปที่ 2.20 แสดงสี่เหลี่ยมครอบส่วนที่ทับกัน ดังเส้นสีน้ำเงิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.21 แสดงการนำสี่เหลี่ยมที่ได้จากรูปที่ 2.20 มาวางในเฟรม Ft
ซึ่งก็จะได้ขนาดและตำแหน่งของวัตถุที่เคลื่อนที่อยู่

เมื่อได้ตำแหน่งและขนาดของวัตถุในเฟรมปัจจุบันแล้วก็สามารถคำนวณหาจุดศูนย์กลางของวัตถุนั้นได้ ซึ่งเมื่อนำมาเทียบกับจุดศูนย์กลางของภาพแล้ว ก็จะได้เวกเตอร์ที่เป็นข้อมูลเพื่อที่ใช้ในการบังคับให้อุปกรณ์เคลื่อนที่ตามวัตถุ เพื่อให้วัตถุนั้นอยู่ในมุมมองของกล้องตลอดเวลา ดังรูปที่ 2.22



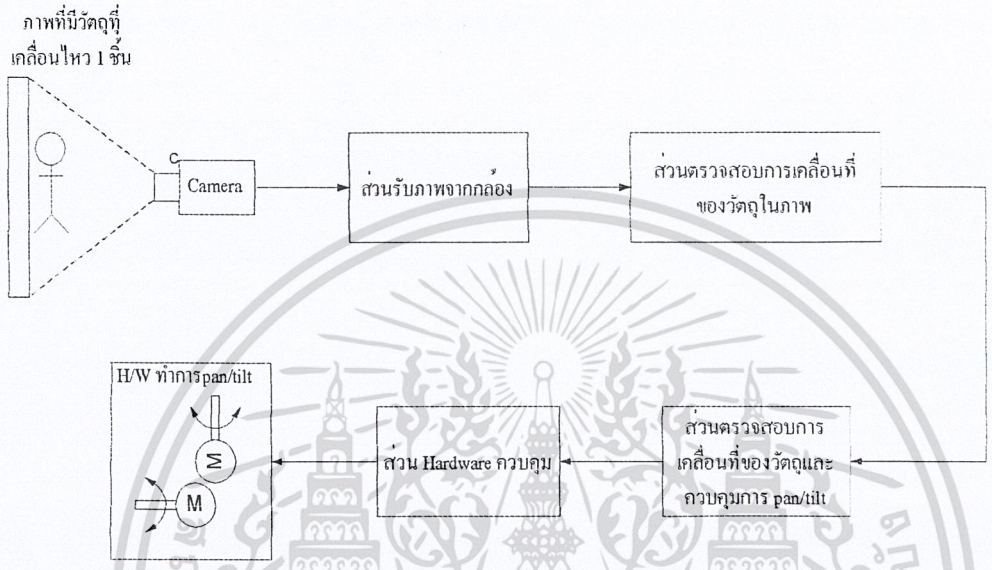
รูปที่ 2.22 แสดงเวกเตอร์ที่ใช้เป็นข้อมูลในการบังคับอุปกรณ์ให้ทำการหมุนเคลื่อนที่ตามวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

โครงสร้างและการออกแบบ

3.1 ภาพรวมของระบบ



รูปที่ 3.1 แสดงภาพรวมของโครงงานกล้องติดตามวัตถุ

จากรูปที่ 3.1 สามารถแบ่งภาพรวมของโครงงานกล้องติดตามวัตถุเป็นส่วนย่อยๆ ได้ ดังนี้

1. ส่วนของกล้อง

2. ส่วนรับภาพจากกล้อง เป็นตัว Capture ภาพเข้ามา เพื่อนำภาพที่ได้ไปใช้ในส่วนถัดไป

3. ส่วนตรวจสอบการเคลื่อนที่ของวัตถุในภาพ จะนำภาพที่ได้จากการ Capture มาทำการตรวจสอบการเคลื่อนที่ของวัตถุ

4. ส่วนตรวจสอบเพื่อบังคับการ pan / tilt จะนำผลที่ได้จากการตรวจสอบการเคลื่อนที่ของวัตถุมาทำการตัดสินใจเพื่อที่จะบังคับให้ระบบควบคุมบังคับการเคลื่อนที่ของกล้อง ทำการ pan และ tilt ตามการเคลื่อนที่ของวัตถุ เพื่อให้วัตถุอยู่ในมุมมองของกล้องตลอดเวลา

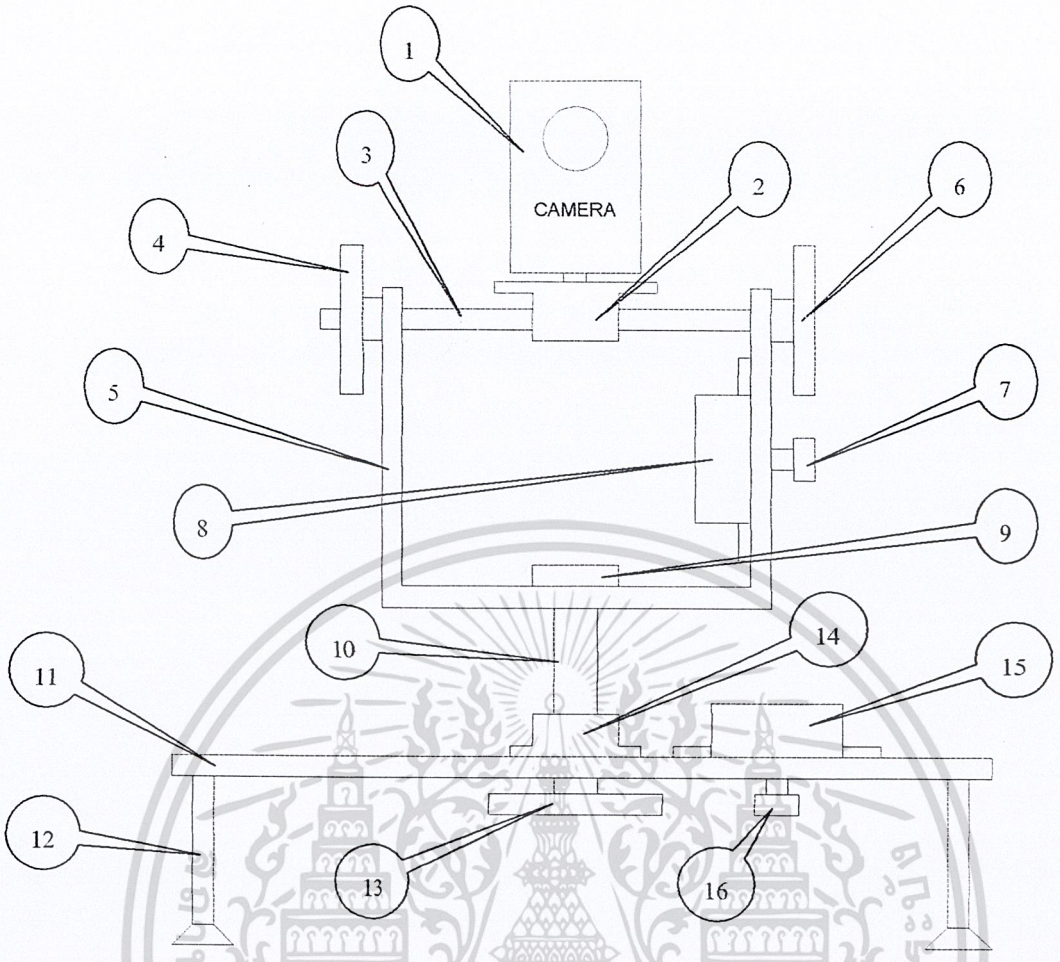
5. ส่วน Hardware ควบคุม เป็นส่วนควบคุมมอเตอร์เพื่อทำให้กล้องเคลื่อนที่ไปตามผลจากส่วนที่ 4

6. ส่วน มอเตอร์ ทำการ pan / tilt ประกอบด้วยมอเตอร์ 2 ตัว

3.2 การออกแบบโครงสร้างของฐานกล้อง

ฐานกล้องสามารถเคลื่อนที่ได้ในลักษณะการ หันซ้าย-ขวา และ ก้ม-เงย ทั้งหมดได้ 4 ทิศทาง การทำงานจะถูกควบคุมโดยอุปกรณ์อิเล็กทรอนิกส์ และโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51 โครงสร้างทั้งหมดของฐานกล้องทำจากสแตนเลส และอะลูมิเนียม ซึ่งมีส่วนประกอบต่างๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงโครงสร้างของฐานกล้อง

ส่วนประกอบต่างๆมีดังนี้

1. กล้องดิจิตอล
2. จุดยึดกล้องดิจิตอล
3. แกนหมุนปรับมุมก้มเงย
4. ตัวยึดแกนหมุน
5. โครงสร้างด้านบน
6. เฟืองแกนหมุนปรับมุมก้มเงย
7. เฟืองมอเตอร์
8. มอเตอร์ปรับหมุนก้มเงย
9. ตัวยึดแกนหมุนกับโครงสร้างด้านบน
10. แกนหมุนซ้ายขวา
11. ฐานสแตนเลส
12. ขาตั้ง
13. เฟืองแกนหมุนซ้ายขวา
14. ตลับลูกปืน

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

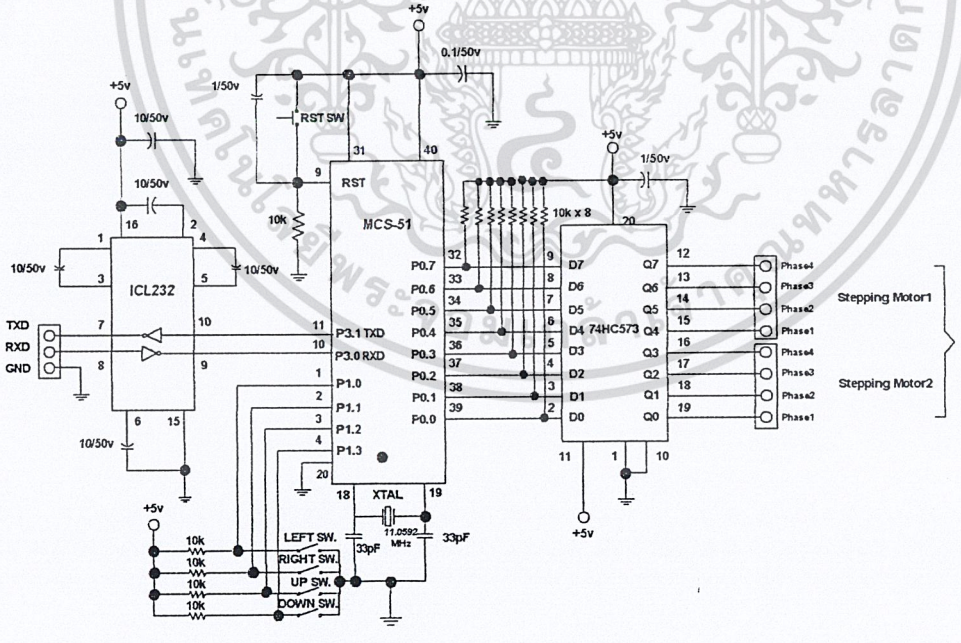
15. มอเตอร์ปรับหมุนซ้ายขวา

16. เฟืองมอเตอร์

เมื่อทำการประกอบชิ้นส่วนต่างๆ เสร็จแล้วก็จะมีลักษณะดังรูปที่ 3.2 ซึ่งแสดงตำแหน่งของอุปกรณ์ต่างๆ ที่ยึดไว้เรียบร้อยแล้ว อุปกรณ์ที่นำมาติดตั้งนั้นได้แก่ กล้องดิจิตอล วงจรควบคุมการทำงาน MCS-51 และวงจรขับสเต็ปปิ้งมอเตอร์ ในการออกแบบนั้นได้ออกแบบให้การติดตั้งอุปกรณ์ต่างๆ มีจุดยึดที่สามารถถอดออกได้สะดวก เพื่อประโยชน์ในการปรับปรุงประสิทธิภาพการทำงานต่อไป

3.3 *การออกแบบวงจรควบคุมการทำงาน*

การออกแบบวงจรควบคุมการทำงานนั้น ประกอบด้วยวงจรของไมโครคอนโทรลเลอร์ MCS-51 ที่มีโปรแกรมควบคุมสั่งงานสเต็ปปิ้งมอเตอร์ให้หมุนก้มเงย และหมุนซ้ายขวา วงจรไมโครคอนโทรลเลอร์ MCS-51 นี้จะเชื่อมต่อกับวงจรบัฟเฟอร์และวงจรไดรเวอร์ ซึ่งทำหน้าที่ขยายกระแสในการควบคุมการทำงานของมอเตอร์ให้มีประสิทธิภาพสูงสุด การควบคุมมอเตอร์นั้นสิ่งสำคัญคือกระแสที่จ่ายให้กับขดลวดของมอเตอร์ต้องคงที่เสมอ เพราะการหมุนแต่ละครั้งนั้นต้องมีความต้องแม่นยำ การออกแบบแหล่งจ่ายที่ไม่มีประสิทธิภาพจะทำให้ค่าตำแหน่งการหมุนผิดพลาดได้ การออกแบบวงจรจ่ายกระแสให้ระบบจึงจำเป็นต้องมีวงจรขยายกระแสที่มีประสิทธิภาพสูง ในการออกแบบโครงงานนี้ได้ออกแบบวงจรเพาเวอร์ซัพพลายโดยมีไอซีเร็กกูเลทเบอร์ 7805 ในการควบคุมแรงดันให้กับวงจรของไมโครคอนโทรลเลอร์ MCS-51 และมีไอซีเร็กกูเลทเบอร์ 7812 ในการควบคุมแรงดันให้กับวงจรไดรเวอร์ ที่ทำหน้าที่ขยายกระแสในการควบคุมการทำงานของมอเตอร์



รูปที่ 3.3 แสดงส่วนไมโครคอนโทรลเลอร์ที่เชื่อมต่อกับวงจรบัฟเฟอร์และพอร์ตอนุกรมของคอมพิวเตอร์

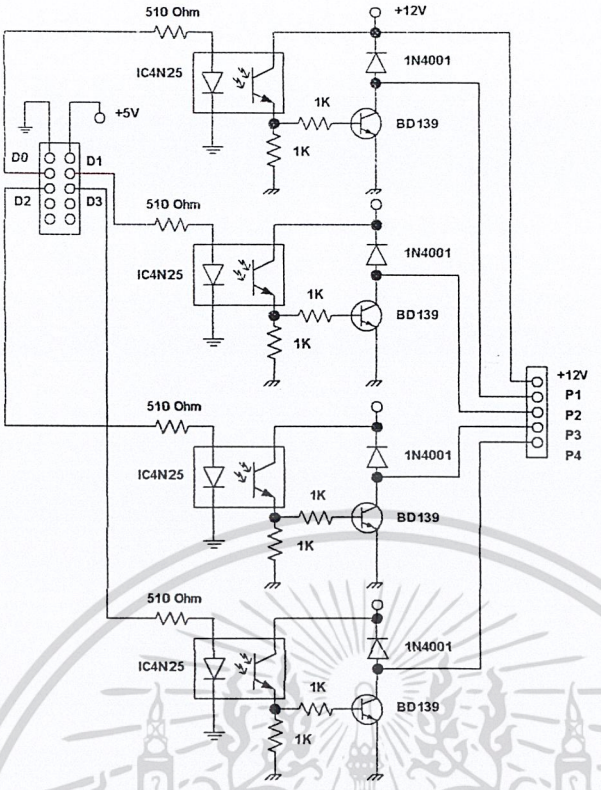
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.3 ตัวไมโครคอนโทรลเลอร์ MCS-51 จะเป็นตัวควบคุมและสั่งงานให้สเต็ปปิ้งมอเตอร์ทำการหมุนก้ม-เงย และหมุนซ้าย-ขวา โดยจะรับข้อมูลการสั่งงานจากวงจรเชื่อมต่อพอร์ตอนุกรมของคอมพิวเตอร์ซึ่งมีไอซี ICL232 เป็นตัวทำหน้าที่ในการแปลงข้อมูลระดับ RS-232 ที่รับจากคอมพิวเตอร์ไปเป็นระดับทีทีแอล ก่อนที่จะถ่ายทอดไปยังไมโครคอนโทรลเลอร์ MCS-51 เพื่อทำการประมวลผลต่อไป (เนื่องจาก MCS-51 ไม่เข้าใจสัญญาณ RS-232)ได้ หน้าที่อีกอย่างหนึ่งของ ICL232 คือแปลงข้อมูลส่งของไมโครคอนโทรลเลอร์ MCS-51 จากระดับทีทีแอลไปเป็นระดับของ RS-232 เพื่อใช้ตรวจสอบการทำงานของไมโครคอนโทรลเลอร์บนเครื่องคอมพิวเตอร์

พอร์ต 1 ของไมโครคอนโทรลเลอร์ใช้ในการรับอินพุตจากสวิทช์ 4 ตัวเพื่อจำกัดการหมุนก้ม-เงยและหมุนซ้าย-ขวา จะเห็นได้ว่า P1.0 – P1.3 จะต่อกับสถานะลอจิก '1' อยู่ตลอดเวลา เมื่อใดก็ตามที่มีอินพุตเข้ามาจากสวิทช์ตัวใดตัวหนึ่ง สถานะที่พอร์ต 1 ของบิตที่ต่อสวิทช์นั้นอยู่จะมีสถานะลอจิก '0' นั่นแสดงว่าไม่สามารถที่จะสั่งงานให้สเต็ปปิ้งมอเตอร์หมุนต่อไปในทิศทางการหมุนเดิมนั้นได้ และจะใช้พอร์ต 0 ของไมโครคอนโทรลเลอร์ MCS-51 ในการส่งข้อมูลผ่านไอซีแลตช์ 74HC573 แล้วส่งต่อไปยังวงจร ไดรเวอร์กระแสสูง ซึ่งแสดงดังรูปที่ 3.4 วงจรไดรเวอร์ประกอบด้วยตัวออปโต้ทรานซิสเตอร์ IC4N25 จำนวน 4 ตัว ตามเฟสของตัวสเต็ปปิ้งมอเตอร์ IC4N25 จะเริ่มทำงานเมื่อมีสัญญาณข้อมูล '1' เข้ามาเป็นผลทำให้ทรานซิสเตอร์ BD139 ที่ต่ออยู่ทำงานขยายกระแสตามไปด้วย ทำให้มีกระแสไปขับสเต็ปปิ้งมอเตอร์ได้โดยวงจรไดรเวอร์นี้สามารถขยายกระแสได้สูงสุด 2 แอมป์ ทั้งนี้ต้องเตรียมแหล่งจ่ายไฟฟ้าให้มีความสามารถในการจ่ายไฟฟ้าเพียงพอด้วย

เมื่อต้องการให้สเต็ปปิ้งมอเตอร์ที่ทำหน้าที่หมุนซ้าย-ขวาทำการหมุน จะต้องส่งข้อมูล '1' ไล่เรียงไปตามลำดับจาก P0.0 ถึง P0.3 แล้ววนกลับมาที่ P0.0 ใหม่ หากต้องการให้สเต็ปปิ้งมอเตอร์กลับทิศทางก็ให้ส่งข้อมูลย้อนกลับ โดยเริ่มจาก P0.3 ก่อนแล้วสิ้นสุดรอบที่ P0.0 แล้ววนกลับมาที่ P0.3 ใหม่ ถ้าต้องการให้สเต็ปปิ้งมอเตอร์ที่ทำหน้าที่หมุนก้ม-เงยทำการหมุน จะต้องส่งข้อมูล '1' ไล่เรียงไปตามลำดับทำนองเดียวกับสเต็ปปิ้งมอเตอร์ที่ทำหน้าที่หมุนซ้าย-ขวา แต่ว่าจะเรียงจาก P0.4 ถึง P0.7 แล้วกลับมาเริ่มใหม่ที่ P0.4 ใหม่อีก การให้สเต็ปปิ้งมอเตอร์หมุนกลับทิศทางทำได้โดยส่งข้อมูลเริ่มจาก P0.7 ถึง P0.4 แล้วกลับมาเริ่มใหม่ที่ P0.7 ใหม่ เมื่อวงจรไดรเวอร์ได้รับข้อมูล "1" ผ่านมาทางไอซี 74HC573 วงจรก็จะทำงาน ทำให้เกิดกระแสไฟฟ้าไหลผ่านขดลวดของมอเตอร์ที่ต่ออยู่ ส่งผลให้เกิดการเคลื่อนที่ของแกนมอเตอร์ขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

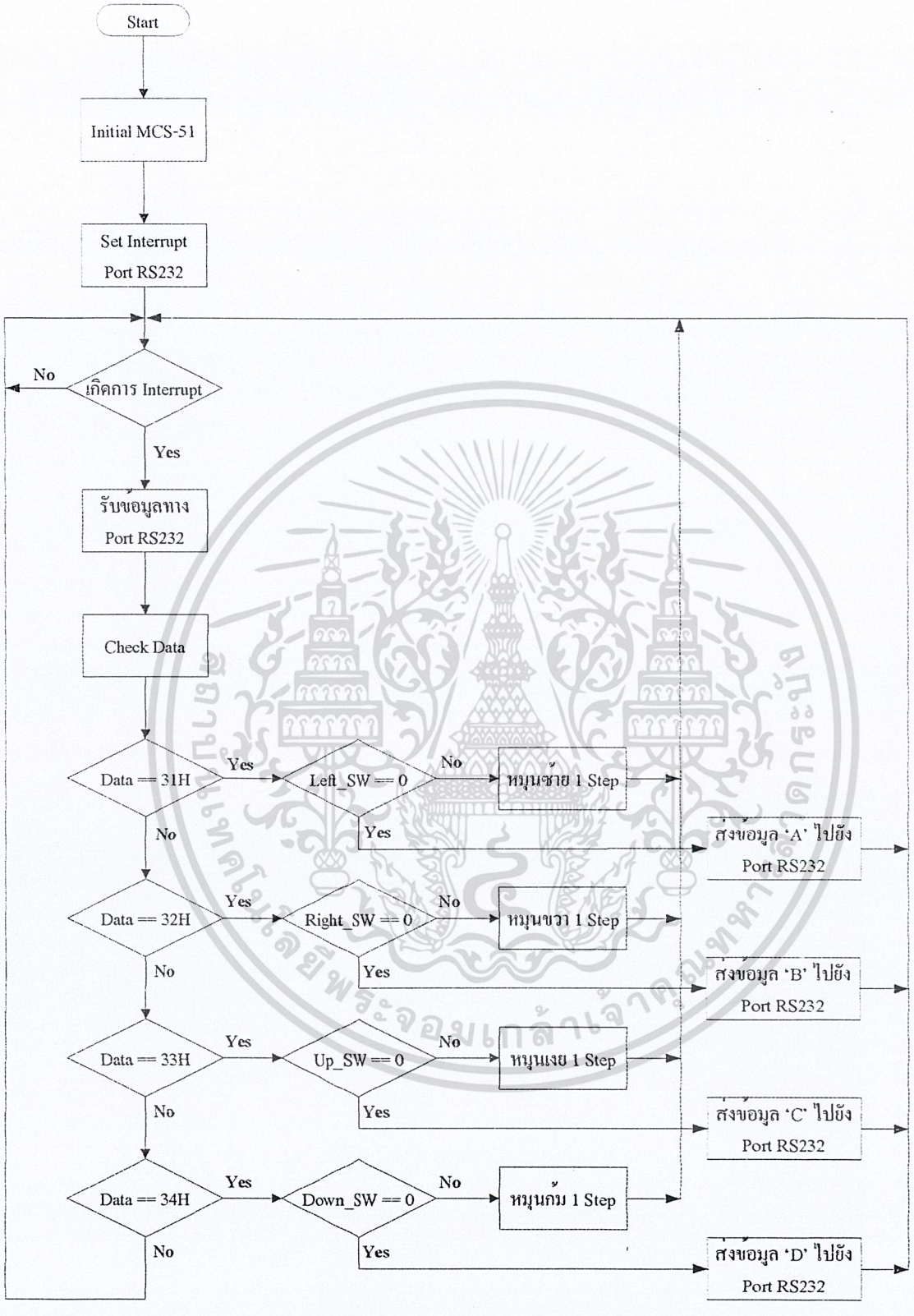


รูปที่ 3.4 แสดงวงจรไดรเวอร์สเต็ปปิ้งมอเตอร์

3.4 การออกแบบโปรแกรมควบคุมสเต็ปปิ้งมอเตอร์ของไมโครคอนโทรลเลอร์ MCS-51

ในการเขียนโปรแกรมกระตุ้นและควบคุมการหมุนของมอเตอร์ให้เคลื่อนที่จะใช้แบบครึ่งสเต็ป (Half Step) เพราะว่าสามารถเพิ่มจำนวนของสเต็ปต่อรอบได้อีกหนึ่งเท่าตัว และแรงบิดที่ได้จากการกระตุ้นแบบนี้จะเพิ่มมากขึ้นอีก เพราะช่วงสเต็ปมีระยะสั้นลง แต่ละสเต็ปเกิดแรงดึงจากขดลวด 2 ขดที่ถูกกระตุ้นพร้อมกัน ความถูกต้องของตำแหน่งก็มีเพิ่มมากขึ้น การเขียนโปรแกรมควบคุมสเต็ปปิ้งมอเตอร์จะใช้หลักการอินเตอร์รัปต์ของพอร์ต RS232 เมื่อเกิดอินเตอร์รัปต์ขึ้นโปรแกรมจะนำข้อมูลที่รับมาทางพอร์ต RS232 มาตรวจสอบ ในที่นี้ ถ้าข้อมูลตัวเลข 1 (31H)เข้ามา 1 ตัวโปรแกรมจะสั่งงานให้หมุน สเต็ปปิ้งมอเตอร์ไปทางซ้าย 1 สเต็ป ถ้าข้อมูลตัวเลข 2 (32H)เข้ามา 1 ตัวโปรแกรมจะสั่งงานให้หมุนสเต็ปปิ้งมอเตอร์ไปทางขวา 1 สเต็ป ถ้าหากเป็นข้อมูลตัวเลข 3 (33H)เข้ามา 1 ตัวโปรแกรมจะสั่งงานให้หมุนสเต็ปปิ้งมอเตอร์เงยขึ้น 1 สเต็ป และถ้าข้อมูลตัวเลข 4 (34H)เข้ามา 1 ตัวโปรแกรมจะสั่งงานให้หมุนสเต็ปปิ้งมอเตอร์ก้มลง 1 สเต็ป สเต็ปปิ้งมอเตอร์จะหมุนไปทางซ้ายได้เมื่อ Left_SW เท่ากับ 1 เมื่อใดก็ตามที่ Left_SW เท่ากับ 0 แสดงว่าสเต็ปปิ้งมอเตอร์ไม่สามารถจะหมุนไปทางซ้ายได้อีก ในการหมุนไปทางขวา ก้ม และเงยก็จะเป็นไปในลักษณะกัน เพียงแต่จะมี Right_SW เป็นตัวควบคุมการหมุนต่อไปได้ของการหมุนขวา Up_SW เป็นตัวควบคุมการหมุนต่อไปได้ของการเงยขึ้น และDown_SW เป็นตัวควบคุมการหมุนต่อไปได้ของการหก้มลง ถ้าหากว่าสเต็ปปิ้งมอเตอร์ไม่สามารถหมุนต่อไปได้ไม่ว่าทางใด ตัวไมโครคอนโทรลเลอร์จะส่งข้อมูลกลับไปยังเครื่องคอมพิวเตอร์ทางพอร์ต RS232 เพื่อบอกว่าไม่สามารถหมุนมอเตอร์ต่อไปได้อีก Flowchart ของโปรแกรมแสดงดังรูปที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

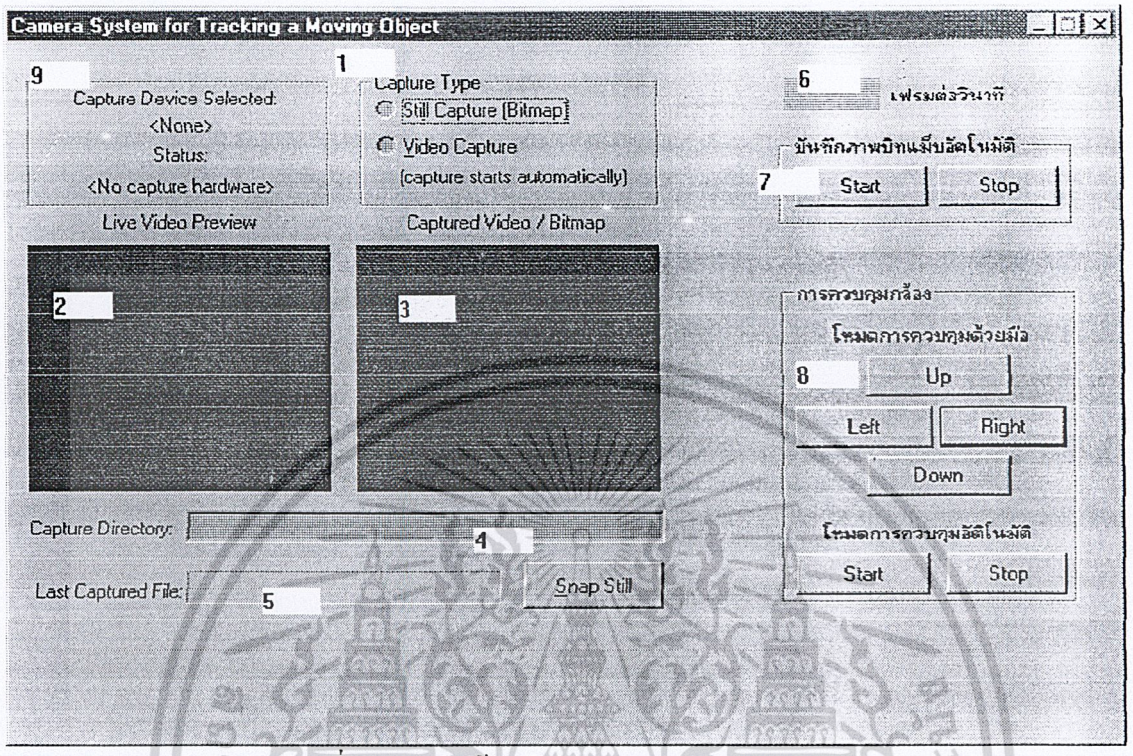


รูปที่ 3.5 แสดง Flowchart ของโปรแกรมควบคุมสเต็ปปิ้งมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 *การออกแบบหน้าจอ*

หน้าจอโปรแกรมของระบบแสดงดังรูปที่ 3.6



รูปที่ 3.6 แสดงหน้าจอโปรแกรมของระบบ

รายละเอียดของหน้าจอมีดังนี้

1. ประเภทของการแคปเจอร์ (Capture) ภาพจากกล้อง ซึ่งจะมีอยู่สองประเภทคือ การแคปเจอร์เป็นภาพนิ่งและการแคปเจอร์เป็นไฟล์วิดีโอ ถ้าเลือกเป็น โหมดของการแคปเจอร์เป็น ไฟล์วิดีโอ จะเริ่มแคปเจอร์ทันที

2. ส่วนแสดงภาพเป็นสัญญาณภาพวิดีโอสดที่ได้ จากกล้อง

3. ส่วนแสดงภาพนิ่งที่ได้จากการถ่ายรูป หรือจากการบันทึกภาพบิทแม็บอัตโนมัติ หรือจากโหมดการควบคุมกล้องอัตโนมัติ

4. ช่องสำหรับป้อนชื่อไดเร็คเทอรี่ที่ใช้เก็บไฟล์ที่ถูกแคปเจอร์

5. ส่วนแสดงชื่อไฟล์ที่ถูกแคปเจอร์ล่าสุด โดยชื่อไฟล์ที่ถูกบันทึกนั้นมีรูปแบบดังนี้ ปี-เดือน-วันที่-ชั่วโมง-นาที-วินาที-1/100 วินาที เช่น 2003-03-18-21-37-57-71, 2003-03-18-21-38-00-48 เป็นต้น

6. ช่องรับจำนวนเฟรมในการแคปเจอร์ในหนึ่งวินาที ซึ่งจะใช้ในการบันทึกภาพบิทแม็บอัตโนมัติ และใช้ในโหมดการควบคุมกล้องอัตโนมัติ

7. ส่วนของการบันทึกภาพบิทแม็บอัตโนมัติ ซึ่งเริ่มทำการแคปเจอร์โดยการกดปุ่มสต้าท ถ้าจะหยุดก็ให้กดปุ่มสต้อป

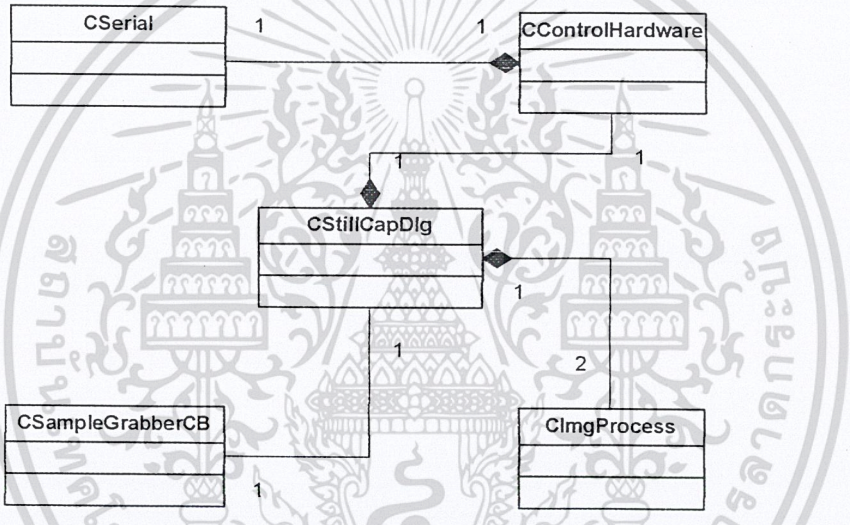
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. ส่วนของการควบคุมกล้อง ประกอบด้วยสองโหมด โหมดแรกคือโหมดการควบคุมด้วยมือ โหมดนี้ประกอบด้วยสี่ปุ่ม เพื่อใช้ในการควบคุมอุปกรณ์ คือปุ่มหมุนไปทางซ้าย ปุ่มหมุนไปทางขวา ปุ่มหมุนขึ้น และปุ่มหมุนลง สำหรับโหมดที่สองคือโหมดการควบคุมกล้องอัตโนมัติ ในโหมดนี้มีอยู่สองปุ่ม ด้วยกันปุ่มแรกเป็นปุ่มเริ่มการควบคุม และปุ่มที่สองเป็นปุ่มหยุดการควบคุม

9. ส่วนแสดงสถานะต่างๆ แบ่งเป็นสองส่วนย่อยดังนี้ ส่วนแรกจะแสดงว่าได้เลือกอุปกรณ์แคปเจอร์อะไร ส่วนที่สองแสดงสถานะของระบบว่ากำลังทำอะไรอยู่เช่น ทำการบันทึกบิทแม็บอัตโนมัติ หรือทำการควบคุมกล้องโหมดอัตโนมัติ หรือว่าทำการบันทึกเป็นไฟล์วิดีโออยู่

3.6 *คลาสไดอะแกรม*

คลาสไดอะแกรมของโปรแกรมแสดงดังรูป 3.7



รูปที่ 3.7 แสดงคลาสไดอะแกรมของโปรแกรม

3.7 ลักษณะของคลาสต่างๆ

3.7.1 คลาส **CSerial** ทำหน้าที่ ติดต่อกับพอร์ตอนุกรม เมธอดที่สำคัญได้แก่ เมธอด Open ทำหน้าที่เปิดพอร์ตคอม 1 หรือ คอม 2 เมื่อต้องการส่งข้อมูลไปยังพอร์ตอนุกรมก็ต้องทำการเปิดพอร์ตก่อน เมื่อทำการเปิดพอร์ตสำเร็จจึงจะเริ่ม ทำการส่งข้อมูลโดย เมธอด SendData และทำการรับข้อมูลโดยใช้เมธอด ReadData สำหรับแอ็คทริบิวท์และเมธอดของ คลาสนี้แสดงดังรูป 3.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CSerial
#m_bOpened : bool
+CSerial() +~CSerial() +IsOpened() : bool +Open(in nPort : int, in bBaud : int) : bool +ReadData(out buffer : void, in limit : int) : int +SendData(in buffer : char, in size : int) : int

รูปที่ 3.8 แสดงลักษณะของ คลาส CSerial

3.7.2 *คลาส* **CControlHardware** ทำหน้าที่ ควบคุมอุปกรณ์บังคับการเคลื่อนที่ของกล้อง เมธอดที่สามารถเรียกใช้ได้คือ เมธอด move โดยสามารถเรียกใช้ได้สองแบบด้วยกัน วิธีแรกมีไว้สำหรับ การควบคุมโหมดการบังคับด้วยมือ สามารถเรียกเมธอดที่มีพารามิเตอร์สองตัว ส่วนวิธีที่สองมีไว้สำหรับการควบคุมโหมดการบังคับอัตโนมัติ โดยเรียกเมธอดที่มีพารามิเตอร์ตัวเดียว พารามิเตอร์ตัวนี้เป็นเวกเตอร์บอกถึงตำแหน่งจุดศูนย์กลางของวัตถุที่เคลื่อนที่อยู่ในภาพ โดยเมธอด move ทั้งสองตัวนี้จะมีลักษณะการทำงานที่เป็นเทรด ลักษณะของคลาสนี้ดังรูปที่ 3.9

CControlHardware
-serial : CSerial -m_port_number : int
+CControlHardware(in port_number : int = 2) +move(in the_vector, in) : void +move(in msg : int, in step : int) : void -start() : void -start_controller() : void -turn_down() : bool -turn_up() : bool -turn_right() : bool -turn_left() : bool

รูปที่ 3.9 แสดงลักษณะของคลาส CControlHardware

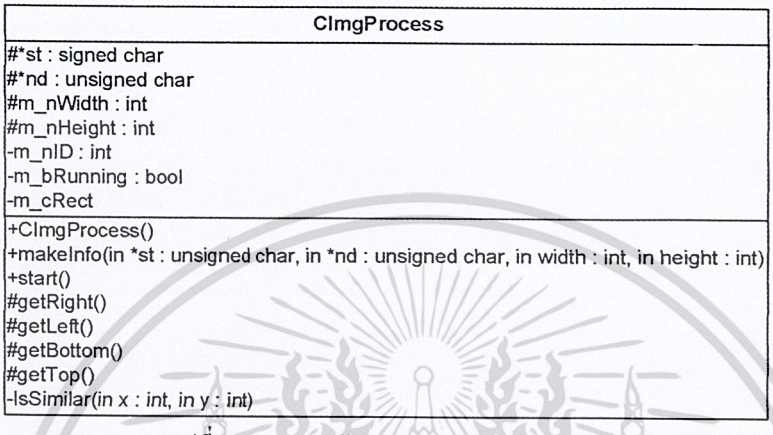
3.7.3 *คลาส* **CSampleGrabberCB** ทำหน้าที่ติดต่อกับกล้อง อ่านภาพ แสดงภาพ และบันทึกเป็นไฟล์บิทแม็บ เมธอดที่ใช้อ่านภาพจากกล้องคือ BufferCB โดยในการอ่านภาพจากกล้องนี้จะมีลักษณะการทำงานที่เป็นเทรด ถ้าบันทึกภาพลงในบัฟเฟอร์เสร็จแล้วก็จะส่งแมสเซจจ์ ไปให้คลาส CStillCapDlg เพื่อเอาภาพในบัฟเฟอร์ไปใช้งานต่อไป ส่วนการแสดงภาพจะใช้เมธอด DisplayCapturedBits และการ บันทึกภาพจะใช้เมธอด CopyBitmap ลักษณะของคลาสนี้แสดงดังรูป 3.10

CSampleGrabberCB
+m_szCapDir +m_szSnappedName +lHeight : long +lWidth : long
+BufferCB(in dblSampleTime : double, in *pBuffer : unsigned char, in lBufferSize : long) +CopyBitmap(in dblSampletime : double, in *pBuffer : unsigned char, in lBufferSize : long) +DisplayCaptureBits(in *pBuffer : unsigned char, in pbih)

รูปที่ 3.10 แสดงลักษณะของคลาส CSampleGrabberCB

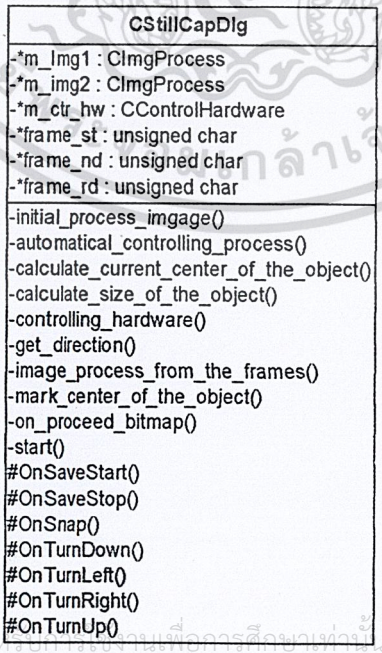
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.4 คลาส **CImgProcess** ทำหน้าที่ประมวลผลภาพตามวิธีที่ได้ประยุกต์ขึ้นมา โดยจะทำการประมวลผลทีละสองภาพ ลักษณะการทำงานจะเป็นเทรด โดยจะเรียกผ่านเมธอด start ถ้าประมวลผลเสร็จแล้วจะส่ง แมสเซจจ์ไปให้ CStillCapDlg นำเอาผลไปใช้งานต่อไป ลักษณะของคลาส CImgProcess แสดงดังรูป 3.11



รูปที่ 3.11 แสดงลักษณะของคลาส CImgProcess

3.7.5 คลาส **CStillCapDlg** ทำหน้าที่เป็นคลาสหลักของโปรแกรมของระบบได้แก่ คอยรับแมสเซจจ์ซึ่งระบุว่าได้ อ่านภาพจากกล้องแล้วเก็บลงในบัฟเฟอร์เรียบร้อยแล้ว จากคลาส CSampleGrabberCB เมื่อได้ภาพมาแล้วก็ทำการส่งให้คลาส CImgProcess ทำการประมวลผล จากนั้นก็คอยรับแมสเซสจ์แจ้งการประมวลผลภาพเสร็จ ที่เหลือก็เป็นส่วนของการนำผลลัพธ์จากการประมวลผลภาพมาวิเคราะห์ เพื่อจะสั่งให้อุปกรณ์ควบคุมกล้องหมุนตามวัตถุที่เคลื่อนที่ต่อไป หน้าที่สำคัญอีกอย่างหนึ่งคือทำหน้าที่ติดต่อกับหน้าจอ



รูปที่ 3.12 แสดงลักษณะบางส่วนของคลาส CStillCapDlg

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง/การทดสอบ

4.1 หาความเร็วเฉลี่ยของสเต็ปมอเตอร์

ในการทดลองนี้จะบอกถึงประสิทธิภาพของฮาร์ดแวร์ที่สร้างขึ้น โดยในการทดลองจะแบ่งเป็นสองส่วนคือการหาความเร็วในการหมุนซ้าย-ขวา และหาความเร็วในการหมุนขึ้น-ลง

4.1.1 หาความเร็วเฉลี่ยในการหมุนซ้าย-ขวา ขั้นตอนการทดลองมีดังนี้

1. กำหนดให้การหมุนของมอเตอร์ หมุนอย่างต่อเนื่อง โดยเขียนโปรแกรมวนลูปสั่งให้มอเตอร์หมุนจำนวน 50 สเต็ป ทำให้ระยะของการเคลื่อนที่ เคลื่อนที่ไป 180 องศา
2. บันทึกเวลาที่เริ่มหมุน
3. บันทึกเวลาที่หยุดหมุน
4. ทำการทดลองซ้ำหลายๆ ครั้งจากนั้นก็คำนวณหาค่าเฉลี่ยของเวลาที่ใช้

ครั้งที่	เวลาเริ่ม	เวลาหยุด	ผลต่างของเวลา
1	24:59:34	25:02:84	0:3:50
2	26:08:37	26:11:89	0:3:52
3	27:26:22	27:29:60	0:3:38
4	28:06:25	28:09:63	0:3:38
5	28:40:53	28:43:93	0:3:34
		รวม	0:17:12

ตารางที่ 4.1 แสดงผลจากการวัดความเร็วในการหมุนซ้าย-ขวา โดยรูปแบบของเวลาคือ นาที:วินาที:หนึ่งส่วนร้อยวินาที

จากการผลการทดลองข้างต้นสามารถหาค่าเฉลี่ยเวลาที่ใช้ในแต่ละครั้งคือ 3 วินาที 424 มิลลิวินาที ดังนั้นสามารถหาเวลาที่ใช้ในการหมุนต่อหนึ่งองศาเป็น 19 มิลลิวินาที หรือเวลาที่ใช้ในการหมุนต่อหนึ่งสเต็ปมอเตอร์เป็น 68.48 มิลลิวินาที

4.1.2 หาความเร็วเฉลี่ยในการหมุนขึ้น-ลง

ขั้นตอนการทดลองมีดังนี้

1. กำหนดให้การหมุนของมอเตอร์ หมุนอย่างต่อเนื่อง โดยเขียนโปรแกรมวนลูปสั่งให้มอเตอร์หมุนจำนวน 13 สเต็ป ทำให้ระยะของการเคลื่อนที่ เคลื่อนที่ไปประมาณ 47 องศา
2. บันทึกเวลาที่เริ่มหมุน
3. บันทึกเวลาที่หยุดหมุน
4. ทำการทดลองซ้ำหลายๆ ครั้งจากนั้นก็คำนวณหาค่าเฉลี่ยของเวลาที่ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ครั้งที่	เวลาเริ่ม	เวลาหยุด	ผลต่างของเวลา
1	43:35:95	43:36:77	0:0:82
2	44:01:30	44:02:11	0:0:81
3	44:25:29	44:26:10	0:0:81
4	44:41:31	44:42:14	0:0:83
5	45:33:07	45:33:90	0:0:83
		รวม	0:4:10

ตารางที่ 4.2 แสดงผลจากการวัดความเร็วในการหมุนขึ้น-ลง โดยรูปแบบของเวลาคือ นาที:วินาที:หนึ่งส่วนร้อยวินาที

จากผลการทดลองข้างต้นสามารถหาค่าเฉลี่ยเวลาที่ใช้ในแต่ละครั้งคือ 820 มิลลิวินาที ดังนั้นสามารถหาเวลาที่ใช้ในการหมุนต่อหนึ่งองศาเป็น 17.45 มิลลิวินาที หรือเวลาที่ใช้ในการหมุนต่อหนึ่งสเต็ปมอเตอร์เป็น 63.08 มิลลิวินาที

4.2 หาอัตราส่วนการเคลื่อนที่ของวัตถุกับจำนวนสเต็ปมอเตอร์

การทดลองนี้จะหาระยะการเคลื่อนที่ของวัตถุเมื่อเทียบกับจำนวนสเต็ปมอเตอร์ เพื่อใช้เป็นค่าในการคำนวณหาจำนวนสเต็ปเพื่อบังคับอุปกรณ์ควบคุมกล้องต่อไป สำหรับการทดลองนี้จะแบ่งเป็นสองส่วนคือ หาอัตราส่วนการเคลื่อนที่ในแนวซ้าย-ขวา และหาอัตราส่วนการเคลื่อนที่ในแนวขึ้น-ลง

4.2.1 *หาอัตราส่วนการเคลื่อนที่ในแนวซ้าย-ขวา* ขั้นตอนการทดลองมีดังนี้

1. เขียนโปรแกรมสั่งให้มอเตอร์หมุนอย่างต่อเนื่องในแนวซ้าย—ขวา
2. บันทึกภาพขณะที่มอเตอร์เริ่มหมุน
3. บันทึกภาพขณะที่มอเตอร์หยุดหมุน
4. หาจุดสังเกตในภาพที่ปรากฏบนภาพทั้งสอง จากนั้นก็บันทึกตำแหน่งที่จุดนั้นปรากฏบนภาพทั้งสองนั้น
5. หาผลต่างของตำแหน่งทั้งสอง
6. ทำการทดลองซ้ำหลายๆ ครั้งแล้วหาค่าเฉลี่ย จำนวนพิกเซลที่เคลื่อนที่ ต่อการหมุนมอเตอร์หนึ่งสเต็ป

ครั้งที่	ตำแหน่งเริ่ม(พิกเซล)	ตำแหน่งหยุด(พิกเซล)	จำนวนสเต็ปที่ใช้	ผลต่างของตำแหน่ง
1	207	57	10	150
2	208	52	10	156
3	208	54	10	154

ตารางที่ 4.3 แสดงผลการบันทึกจากการทดลอง หาอัตราส่วนการเคลื่อนที่ในแนวซ้าย-ขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4	174	34	8	146
5	176	32	8	144
6	160	44	8	116
7	159	88	5	71
8	160	88	5	72
9	159	87	5	72
		รวม	69	1081

ตารางที่ 4.3 (ต่อ) แสดงผลการบันทึกจากการทดลอง หาอัตราส่วนการเคลื่อนที่ในแนวซ้าย-ขวา

จากผลการทดลองข้างต้นสามารถหาอัตราส่วนเฉลี่ยของการเคลื่อนที่ของวัตถุกับจำนวนสเต็ปมอเตอร์หนึ่งสเต็ปคือ 15.667 พิกเซล

4.2.2 หาอัตราส่วนการเคลื่อนที่ในแนวขึ้น-ลง ขั้นตอนการทดลองมีดังนี้

1. เขียนโปรแกรมสั่งให้มอเตอร์หมุนอย่างต่อเนื่องในแนวขึ้น-ลง
2. บันทึกภาพขณะที่มอเตอร์เริ่มหมุน
3. บันทึกภาพขณะที่มอเตอร์หยุดหมุน
4. หาจุดสังเกตในภาพที่ปรากฏบนภาพทั้งสอง จากนั้นก็บันทึกตำแหน่งที่จุดนั้นปรากฏบนภาพทั้งสองนั้น
5. หาผลต่างของตำแหน่งทั้งสอง
6. ทำการทดลองซ้ำหลายๆ ครั้งแล้วหาค่าเฉลี่ย จำนวนพิกเซลที่เคลื่อนที่ ต่อการหมุนมอเตอร์หนึ่งสเต็ป

ครั้งที่	ตำแหน่งเริ่ม(พิกเซล)	ตำแหน่งหยุด(พิกเซล)	จำนวนสเต็ปที่ใช้	ผลต่างของตำแหน่ง
1	301	145	10	156
2	303	148	10	155
3	243	81	10	162
4	231	83	8	148
5	230	81	8	150
6	232	79	8	152
7	181	112	5	69
8	179	108	5	71
9	180	107	5	73
		รวม	69	1136

ตารางที่ 4.4 แสดงผลการบันทึกจากการทดลอง หาอัตราส่วนการเคลื่อนที่ในแนวขึ้น-ลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลการทดลองข้างต้นสามารถหาอัตราส่วนเฉลี่ยของการเคลื่อนที่ของวัตถุกับจำนวนสเต็ปมอเตอร์หนึ่งสเต็ปคือ 16.46 พิกเซล

จากผลการทดลองทั้งการหาอัตราส่วนของการเคลื่อนที่ในแนวขึ้น-ลง และซ้ายขวา ได้ค่าเฉลี่ยเป็น 15.67 และ 16.46 พิกเซลต่อหนึ่งสเต็ป ดังนั้นเมื่อเราได้เวกเตอร์ที่บอกว่าวัตถุที่เคลื่อนที่อยู่นั้นอยู่ห่างจากศูนย์กลางของภาพแล้ว ซึ่งจะบอกเป็นจำนวนพิกเซลจากนั้นก็นำมาแปลงเป็นจำนวนสเต็ปมอเตอร์ เพื่อที่จะสั่งมอเตอร์หมุนเพื่อให้วัตถุนั้นอยู่ในมุมมองของกล้องตลอดเวลา

4.3 *หาเวลาเฉลี่ยที่ใช้ในการประมวลผลภาพและสั่งควบคุมกล้อง*

เนื่องจากในโหมดการควบคุมกล้องอัตโนมัติ มีการรอรับภาพ 3 ภาพเพื่อนำมาประมวลผล จากนั้นก็สั่งให้อุปกรณ์กล้องให้หมุนตามวัตถุนั้น แล้วก็ทำการเริ่มรอรับภาพอีกที ดังนั้นในการทดลองนี้จะหาค่าเฉลี่ยตั้งแต่เริ่มรอรับภาพแรกไปจนสั่งให้กล้องหมุนเสร็จ สำหรับการทดลองนี้จะอ่านภาพจากกล้องด้วยความเร็วที่ 10 เฟรมต่อวินาที โดยขั้นตอนการทดลองมีดังนี้

1. บันทึกเวลาในขณะที่ มีการสั่งให้กล้องหมุนเสร็จในแต่ละครั้ง
2. พยายามให้วัตถุที่ใช้ในการทดลองเคลื่อนที่ตลอดเวลา
3. หาผลต่างของเวลาปัจจุบันกับเวลาก่อนหน้า
4. ทำการทดลองซ้ำหลายครั้ง
5. หาค่าเฉลี่ยของผลต่างของเวลานั้น

ครั้งที่	เวลาที่จับได้(นาที:วินาที:หนึ่งส่วนร้อยวินาที)	ผลต่างของเวลา (วินาที)
0	06:41:73	-
1	06:42:25	0.52
2	06:42:70	0.45
3	06:43:24	0.54
4	06:44:38	1.14
5	06:44:93	0.55
6	06:45:54	0.61
7	06:46:05	0.51
8	06:46:59	0.54
9	06:47:16	0.57
10	06:47:65	0.49
	รวม	5.92

ตารางที่ 4.5 แสดงผลการบันทึกจากการทดลองการหาเวลาเฉลี่ยที่ใช้ในการประมวลผลภาพและสั่งควบคุมกล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลการทดลองสามารถหาค่าเฉลี่ยของเวลาที่ใช้ในการประมวลผลภาพ และการสั่งการควบคุมกล้องคือ5.92/10 = 0.592 วินาที จากผลข้างต้นจะเห็นว่าค่าที่ได้ไม่คงที่ เนื่องจากระยะที่วัตถุเคลื่อนที่นั้นไม่เท่ากัน ทำให้จำนวนสเต็ปที่สั่งให้มอเตอร์หมุนไม่เท่ากัน ถ้าตำแหน่งของวัตถุอยู่ที่ขอบของภาพก็จะใช้เวลานาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และสรุป

5.1 บทวิจารณ์

จากการทดลองทำให้พบข้อจำกัดต่างๆ ของโครงงานกล้องติดตามวัตถุดังต่อไปนี้

1. ตัวสเต็ปปิ้งมอเตอร์ที่นำมาใช้ ในการหมุน 1 สเต็ป จะสามารถหมุนไปได้ 3.6 องศา จึงทำให้ความละเอียดในการหมุนติดตามวัตถุมีความละเอียดไม่มาก ซึ่งอาจจะทำให้การติดตามวัตถุมีความ คลาดเคลื่อนได้
2. เนื่องจากภาพที่สามารถอ่านจากกล้องได้ด้วยความเร็วสูงสุดเพียง 10 เฟรมต่อวินาทีเท่านั้น ทำให้สามารถจับภาพได้เฉพาะวัตถุที่เคลื่อนที่ด้วยความเร็วต่ำ เท่านั้น ถ้าวัตถุเคลื่อนที่ด้วยความเร็วสูง วัตถุนั้นอาจไม่ปรากฏบนภาพที่อ่านมาจากกล้อง จึงไม่สามารถตรวจจับได้
3. เนื่องจากอุปกรณ์ควบคุมการหมุนของกล้องมีความไว ทำให้ภาพที่อ่านจากกล้อง มีความเบลอมากเนื่องจากการเคลื่อนกล้อง ดังนั้นถ้าจะพัฒนาให้สามารถเคลื่อนกล้องไปด้วย แล้วนำภาพมาประมวลผลพร้อมๆ กันจะต้องแก้ไขความเบลอตามทฤษฎีของการประมวลผลภาพ แต่วิธีนั้นอาจเป็นไปได้ยากสำหรับระบบที่เป็น เรียลไทม์ เพราะการแก้ความเบลอของภาพให้เป็นภาพปกตินั้นจะใช้เวลาในการประมวลผลมาก
4. เนื่องจากอุปกรณ์ควบคุมการหมุนของกล้องจะมีการสั่นด้วย ทำให้ต้องรอเวลาให้กล้องหยุดนิ่งเสียก่อน จะนำภาพไปประมวลผลต่อไป ระบบจึงทำงานช้าลงไปอีก ทางแก้ไขควรจะปรับปรุงอุปกรณ์กล้องไม่ให้สั่นในขณะที่กำลังหมุน

5.2 สรุป

โครงงานกล้องติดตามวัตถุ เป็นโครงงานเพื่อเป็นพื้นฐานทางระบบการมองเห็นของหุ่นยนต์ โดยระบบการมองนี้มีทั้ง*ส่วนฮาร์ดแวร์*และซอฟแวร์ ทางด้านฮาร์ดแวร์ประกอบด้วยตัวกล้องที่คอยรับภาพเพื่อเข้ามาประมวลผล และอุปกรณ์ควบคุมกล้อง สามารถควบคุมกล้องให้หมุนซ้าย-ขวา และขึ้น-ลงได้ โดยการหมุนซ้าย-ขวานั้นสามารถหมุนในระยะ 180 องศาและใช้เวลาในการหมุน 180 องศาเท่ากับ 3.424 วินาทีโดยเฉลี่ย ส่วนการก้ม-เงยสามารถหมุนได้ในระยะ 45 องศาและใช้เวลาในการหมุนตลอด 45 องศานั้นเป็น 0.82 วินาที ส่วนทางด้านซอฟแวร์ สามารถรับภาพจากกล้องเข้ามาประมวลผลตามวิธีที่ประยุกต์ขึ้นมา สามารถตรวจสอบการเคลื่อนที่ของวัตถุ และบังคับให้อุปกรณ์ควบคุมกล้องทำการหมุนซ้าย-ขวา และก้ม-เงยตามการเคลื่อนที่ของวัตถุ ทำให้วัตถุปรากฏอยู่ในมุมมองของกล้องตลอดเวลา แต่ถ้าวัตถุที่ปรากฏอยู่ในมุมมองมีมากกว่าหนึ่งชิ้น ระบบจะทำการคำนวณหาเวกเตอร์ของการเคลื่อนที่*รวม*และบังคับให้อุปกรณ์ควบคุมกล้องหมุนกล้องไปตามวัตถุนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ข้อเสนอแนะสำหรับการพัฒนาในอนาคต

1. เพิ่มความสามารถในการหมุนของฐานกล้องให้สามารถควบคุมการหมุนและติดตามวัตถุในระยะซ้าย-ขวา ได้ 0-360 องศา เพื่อจะได้ติดตามวัตถุที่สนใจได้ตลอดเวลา
2. เพิ่มความแม่นยำในการหมุนติดตามวัตถุของฐานกล้องให้มากขึ้น โดยใช้สเต็ปปิ้งมอเตอร์ที่มีขนาดองศาของการหมุนต่อ 1 สเต็ปมอเตอร์น้อยกว่าปัจจุบัน
3. เพิ่มความสามารถในการติดตามวัตถุ ให้สามารถติดตามวัตถุที่เคลื่อนที่เร็วๆ โดยการใช้กล้องวีดิโอที่มีคุณภาพดีขึ้น และเพิ่มอัตราการ Sampling ภาพให้เร็วขึ้น เพื่อที่จะได้ติดตามวัตถุที่เคลื่อนที่เร็วๆ ได้
4. เพิ่มความสามารถให้สามารถติดตามวัตถุที่เคลื่อนที่ที่เราสนใจ ในสภาวะแวดล้อมที่มีวัตถุเคลื่อนที่มากกว่า 1 วัตถุในมุมมองเดียวกัน ซึ่งต้องวิจัยและหาเทคนิคที่เหมาะสมและถูกต้องที่สุดต่อไป
5. เพิ่มความสามารถในการสั่งงานควบคุมการติดตามวัตถุ ให้สามารถควบคุมการติดตามวัตถุผ่านทางระบบเน็ทเวิร์ก หรือควบคุมผ่านทางอินเตอร์เน็ท หรืออาจจะควบคุมผ่านทางอุปกรณ์ไร้สายต่างๆ ซึ่งต้องวิจัยและหาเทคนิคที่เหมาะสมและถูกต้องที่สุดต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. ชัยวัฒน์ ลิ้มพรจิตรวิไล และ วรพจน์ กรแก้ววัฒนกุล, ไมโครคอนโทรลเลอร์ MCS-51, บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด, 2540
2. Keshab K. Parhi and Takao Nishitani, Digital Signal Processing for Multimedia Systems, Marcel Dekker, Inc
3. Yong-Sheng Chen, Yi-Ping Hung, Fast Block Matching Algorithm Based on the Winner-Update Strategy, IEEE Transactions on Image Processing , VOL 10, NO.8, AUGUST 2001
4. Microsoft DirectX 8.1 Document for C++



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้