

ระบบรวมคะแนนข้อสอบอัตโนมัติแบบอัตโนมัติ
AUTOMATIC SCORING OF ANSWER SHEET

โดย

นายนทีธร ชูสีกราน
นายบุรฉัตร จันทรสุต
นายปิยะดล ถมวัฒน์ศิลป์

ปฏิญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

ระบบรวมคะแนนข้อสอบอัตโนมัติแบบอัตโนมัติ
AUTOMATIC SCORING OF ANSWER SHEET

โดย

นายนทีธร	ชูลีกราน	57010647
นายบุรฉัตร	จันทรสูทร	57010706
นายปิยะดล	ถมวัฒน์ศิลป์	57010785

อาจารย์ที่ปรึกษา

ดร.เวจิต ภาคย์พิสุทธิ

ศ.ดร.พรชัย ทรัพย์นิธิ

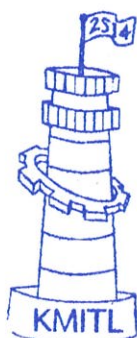
ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560



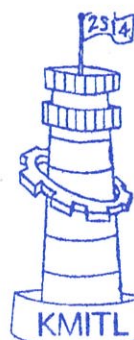
ผ่านการตรวจรูปเล่มแล้ว

(.....)

อาจารย์ที่ปรึกษา

15/5/61

วิศวกรรมโทรคมนาคม
Telecommunications Engineering



ผ่านการตรวจชิ้นงานแล้ว

(.....)

กรรมการผู้ตรวจชิ้นงาน

18/5/61

วิศวกรรมโทรคมนาคม
Telecommunications Engineering

ปริญญาโทปีการศึกษา 2560

ภาควิชาวิศวกรรมโทรคมนาคม

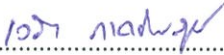
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบรวมคะแนนข้อสอบอัตโนมัติแบบอัตโนมัติ

AUTOMATIC SCORING OF ANSWER SHEET

ผู้จัดทำ

- | | | |
|---------------|--------------|----------|
| 1. นายนทีธร | ชวลีกราน | 57010647 |
| 2. นายบุรฉัตร | จันทร์สูตร | 57010706 |
| 3. นายปิยะดล | ถมวัฒน์ศิลป์ | 57010785 |

.....  อาจารย์ที่ปรึกษา
ดร.เวธิต ภาคย์พิสุทธิ์

.....  อาจารย์ที่ปรึกษาร่วม
ศ.ดร.พรชัย ทรัพย์นิธิ

กิตติกรรมประกาศ

ปริญญาบัตรนี้มีเนื้อหาเกี่ยวกับระบบรวมคะแนนข้อสอบแบบอัตนัย จะไม่สำเร็จ
ลุล่วงได้ถ้าหากปราศจากความกรุณา และความช่วยเหลืออย่างสูงจากอาจารย์ที่ปรึกษาหลัก
ดร.เวธิต ภาคย์พิสุทธิ อาจารย์ที่ปรึกษาร่วม ศ.ดร.พรชัย ทรัพย์นิธิ และนักศึกษาปริญญาโทที่ได้ให้
คำปรึกษา คำแนะนำ และแนวทางการแก้ไขจุดบกพร่องของโครงการในกรณีที่พบปัญหาระหว่าง
การจัดทำ รวมถึงการสนับสนุนสถานที่ และเครื่องมืออุปกรณ์ต่าง ๆ ในช่วงการจัดทำโครงการที่
ผ่านมา

ขอขอบคุณ บิดา มารดา เพื่อนนักศึกษา ตลอดจนผู้ที่เกี่ยวข้องที่ไม่ได้กล่าวไว้ข้างต้นที่
ได้ให้กำลังใจและมีส่วนช่วยเหลือให้โครงการนี้สำเร็จลุล่วงได้ ทางผู้จัดทำหวังเป็นอย่างสูงว่า
โครงการนี้จะมีประโยชน์ต่อผู้ที่สนใจไม่มากก็น้อย

นายนิธิร ชุสิกราน

นายบุรฉัตร จันทรสุต

นายปิยะดล ถมวัฒน์ศิลป์

ผู้จัดทำ

ระบบรวมคะแนนข้อสอบอัตโนมัติ
AUTOMATIC SCORING OF ANSWER SHEET

โดย	นายนทีธร ชูลีกราน	57010647
	นายบุรฉัตร จันทรสุต	57010706
	นายปิยะตล ถมวัฒน์ศิลป์	57010785

อาจารย์ที่ปรึกษา ดร.เวธิต ภาคย์พิสุทธิ์

อาจารย์ที่ปรึกษา ศ.ดร.พรชัย ทรัพย์นิธิ

บทคัดย่อ

ปริญญานิพนธ์นี้เป็นการออกแบบโปรแกรมรวมคะแนนข้อสอบแบบอัตโนมัติโดยจัดทำชุดเลื่อนตัวกล้องสองแกน ที่ถูกควบคุมด้วยไมโครคอนโทรลเลอร์ (Microcontroller) เพื่อปรับตำแหน่งกล้องให้ครอบคลุมข้อมูลที่เป็นตัวเลขทั้งหมด กระบวนการทำงานของระบบเริ่มจากโปรแกรมรับภาพจากกล้องเข้าสู่ Convolutional Neural Network หลังจากการประมวลผลเสร็จสิ้นจะส่งต่อการทำงานไปยังไมโครคอนโทรลเลอร์ เพื่อเปลี่ยนชุดกระดาษคำตอบ

ABSTRACT

This thesis aims to design an automatic scoring of answer sheet by using the two-axis camera slider which is controlled by microcontroller for adjustment the camera location to cover all number. The operation system starts from the program receives an image from camera. Then, the program will crop the image and classify the handwritten number with CNN. After finishing the process, the program sends command to microcontroller for change answer sheet.

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	III
สารบัญรูป	V
สารบัญตาราง	IX
บทที่ 1	
บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	2
บทที่ 2	
ทฤษฎีและหลักการที่เกี่ยวข้อง	3
2.1 หลักการเกี่ยวกับฮาร์ดแวร์และระบบควบคุม	3
2.2 หลักการเกี่ยวกับการประมวลผลภาพ	12
2.3 หลักการที่ใช้ในกระบวนการเรียนรู้และจดจำ	18
บทที่ 3	
การออกแบบและการจัดทำปริญญานิพนธ์	33
3.1 การออกแบบ	33
3.2 เครื่องมือที่ใช้ในการทดลอง	68
3.3 การจัดเก็บผลการทดลอง	78

สารบัญ (ต่อ)

	หน้า
บทที่ 4	
ผลการทดลอง	83
4.1 การประกอบเครื่องเปลี่ยนชุดข้อสอบ	83
4.2 ส่วนชุดเลื่อนตัวกลิ้งสองแกนและซอฟต์แวร์ควบคุม	84
4.3 ส่วนชุดเปลี่ยนชุดข้อสอบและซอฟต์แวร์ควบคุม	87
4.4 ส่วนโปรแกรมประมวลผลภาพ	90
4.5 ส่วนโปรแกรมจำแนกตัวเลขด้วยกระบวนการเรียนรู้และจดจำแบบ CNN	95
4.6 ส่วนโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้	100
บทที่ 5	
สรุปผลและข้อเสนอแนะ	108
5.1 สรุปผล	108
5.2 ข้อเสนอแนะ	108
บรรณานุกรม	109
ภาคผนวก	

สารบัญรูป

รูปที่	หน้า	
2.1	กฎมอเตอร์	3
2.2	มอเตอร์กระแสตรง	4
2.3	ลักษณะการพันขดลวดของ STEPPING MOTOR แบบไบโพลาร์	5
2.4	ลักษณะการพันขดลวดของ STEPPING MOTOR แบบยูนิโพลาร์	6
2.5	STEPPING MOTOR	6
2.6	ULTRASONIC SENSOR MODULE	7
2.7	หลักการทำงานของ ULTRASONIC SENSOR MODULE	7
2.8	การส่ง และรับสัญญาณของ ULTRASONIC SENSOR MODULE	8
2.9	ระบบพิกัดคาร์ทีเซียน	9
2.10	การจัดวางตำแหน่งของ STEPPING MOTOR และสายพาน	10
2.11	ความสัมพันธ์ระหว่างทิศทางการหมุนกับตัวแปร ΔA และ ΔB	11
2.12	ระยะที่มอเตอร์ A หมุนได้	11
2.13	ระยะที่มอเตอร์ B หมุนได้	12
2.14	การเปรียบเทียบต้นฉบับกับภาพที่ผ่านตัวกรอง BILATERAL	12
2.15	ภาพตัวเลขลายมือ	14
2.16	กราฟฮิสโตแกรมความเข้มพิกเซลของรูปที่ 2.15	14
2.17	การกำหนดค่าเทรชโฮลด์ทำให้เกิดการแบ่งส่วนของภาพ	15
2.18	ผลจากการนำรูปที่ 2.16 เข้าสู่กระบวนการเทรชโฮลด์	15
2.19	กระบวนการหาคอนทัวร์ด้วยวิธี MOOR NEIGHBOR TRACING M	16
2.20	แบบจำลองโครงข่ายประสาทเทียม 1 หน่วย	19
2.21	กราฟผลตอบสนองของ THRESHOLD FUNCTION	20
2.22	กราฟผลตอบสนองของ SIGMOID FUNCTION	20
2.23	กราฟผลตอบสนองของ HYPERBOLIC TANGENT FUNCTION	21
2.24	กราฟผลตอบสนองของ RESTRICTED LINEAR UNIT FUNCTION	22

สารบัญรูป(ต่อ)

รูปที่	หน้า	
2.25	สถาปัตยกรรมโครงข่ายประสาทเทียมเพอร์เซพตรอนแบบหลายชั้น	23
2.26	เปรียบเทียบผลกระทบจากการปรับอัตราการเรียนรู้ (LEARNING RATE)	26
2.27	ภาพรวมการทำงานของ CONVOLUTIONAL NEURAL NETWORK	26
2.28	ตัวแปรตำแหน่งภาพอินพุตและตัวกรองถ่วงน้ำหนัก	28
2.29	กระบวนการใน CONVOLUTION LAYER	28
2.30	กระบวนการทำงานในชั้น POOLING LAYER	29
2.31	การลดอัตราสุ่มแบบ AVERAGE POOLING	30
2.32	การทำงานใน FULLY-CONNECTED LAYER	31
3.1	บล็อกไดอะแกรมของระบบรวมคะแนนข้อสอบแบบอัตโนมัติ	33
3.2	บล็อกไดอะแกรมของโปรแกรม	34
3.3	ลักษณะการขึ้นโครงอลูมิเนียมโปรไฟล์ที่ออกแบบไว้	35
3.4	การประกอบอุปกรณ์ชุดเลื่อนตัวกลิ้งสองแกนเคลื่อนที่ในแนวแกน X ที่ออกแบบไว้	35
3.5	การประกอบอุปกรณ์ชุดเลื่อนกลิ้งสองแกนเคลื่อนที่ในแนวแกน Y	36
3.6	ลักษณะการรวมอุปกรณ์ชุดเลื่อนกลิ้งสองแกนเคลื่อนที่ในแนวแกน X และ Y	36
3.7	การติดตั้งชิ้นส่วน IDLER	36
3.8	การติดตั้ง MOTOR	37
3.9	การยึดสายพานกับชิ้นส่วน X CARRIAGE	37
3.10	การวางตำแหน่งสายพานผ่านอุปกรณ์ และชิ้นส่วนต่าง ๆ ที่ออกแบบไว้	37
3.11	รายละเอียดช่องเชื่อมต่อของ A4988 STEPPER DRIVER	38
3.12	การเชื่อมต่อระหว่าง Arduino A4988 STEPPER DRIVER และ Stepping Motor ชุด A	39

สารบัญรูป(ต่อ)

รูปที่		หน้า
3.13	การเชื่อมต่อระหว่าง ARDUINO A4988 STEPPER DRIVER และ STEPPING MOTOR ชุด B	40
3.14	รายละเอียดช่องเชื่อมต่อของ ENDSTOP SWITCH	40
3.15	ผังงานการทำงานของโปรแกรมเลื่อนตัวกัล้องสองแกน	41
3.16	ผังงานการทำงานของโปรแกรมตั้งค่าเริ่มต้นของชุดเลื่อนตัวกัล้องสองแกน	44
3.17	การประกอบด้วยกถาตรงข้อสอบ	45
3.18	การประกอบกถาตรงข้อสอบ	46
3.19	การยัดกถาตรงข้อสอบกับด้วยกถาตรงข้อสอบ	46
3.20	การประกอบเพลาตรง	46
3.21	การประกอบ Z MOTOR	47
3.22	ลักษณะการติดตั้งชิ้นส่วน Z MOTOR Z SHAFT CLAMP และฐานรองข้อสอบ	47
3.23	การติดตั้งชุด LEAD SCREW	48
3.24	ตัวอย่างการเชื่อมต่อสายระหว่าง ARDUINO A4988 STEPPER DRIVER STEPPING MOTOR ULTRASONICS SENSOR MODULE และ ENDSTOP SWITCH	49
3.25	ตัวอย่างการเชื่อมต่อสายระหว่าง ARDUINO IC L298N และ DC MOTOR	50
3.26	ผังงานการทำงานของฟังก์ชันเลื่อนฐานรองชุดข้อสอบลง	51
3.27	ผังงานการทำงานของฟังก์ชันเลื่อนฐานรองชุดข้อสอบขึ้น	52
3.28	ผังงานการทำงานของเปลี่ยนชุดข้อสอบ	53
3.29	แผนผังการทำงานของฟังก์ชันตัดกรอบตัวเลข	55
3.30	ผลของโปรแกรมเมื่อภาพจากกล้องตรงกับภาพต้นแบบ	55
3.31	ผลของโปรแกรมเมื่อภาพจากกล้องไม่ตรงกับภาพ	56
3.32	ผลของกระบวนการ PERSPECTIVE TRANSFORM	56
3.33	ผลของการตัดกรอบช่องกรอกคะแนนและแปลงสีภาพเป็นภาพขาวเทา	57

สารบัญรูป(ต่อ)

รูปที่		หน้า
3.34	ภาพหลังจากเข้าสู่ BILATERAL FILTER	57
3.35	ภาพหลังการกระบวนกร THRESHOLDING	58
3.36	ภาพหลังจากการ FIND CONTOURS	58
3.37	แผนผังการทำงานของฟังก์ชันลำดับค่าประจำหลัก	59
3.38	ผลการเรียงลำดับจากค่าพิกัด x	59
3.39	ผลการจำแนกตัวเลขจากลายมือ	60
3.40	การทำงานในลำดับที่ 3	60
3.41	ลำดับการทำงานของฟังก์ชันโครงสร้างโมเดล	61
3.42	แผนผังการทำงานของโปรแกรม GUI	63
3.43	ภาพจำลองหน้าต่างเริ่มต้น	64
3.44	ภาพจำลองหน้าต่างเลื่อนตัวกลิ้ง	65
3.45	ภาพจำลองหน้าต่างเลื่อนตัวกลิ้ง	65
3.46	ภาพจำลองหน้าต่างยืนยันการลากกรอบ	66
3.47	ภาพจำลองหน้าต่างยืนยันชื่อชุดข้อสอบ	66
3.48	ภาพจำลองหน้าต่างตรวจจับตัวเลข	67
3.49	ชิ้นส่วน X CARRIAGE	71
3.50	ชิ้นส่วน XY JOINER	71
3.51	ชิ้นส่วน XY Idler	72
5.52	ชิ้นส่วน X CARRIAGE	72
5.53	ชิ้นส่วน Z NUT MOUNT	72
3.54	ชิ้นส่วน Z MOTOR	72
3.55	ชิ้นส่วน Z SHAFT CLAMP	73
3.56	คอมพิวเตอร์ ASUS TRANSFORMER BOOK FLIP TP550LD	76
3.57	WEB CAMERA OKER FULL HD AUTO FOCUS HD-386	77
3.58	การทดลองความแม่นยำของการเคลื่อนที่ที่ตำแหน่งปลายทาง	78

สารบัญญรูป(ต่อ)

รูปที่	หน้า
3.59 SERIAL MONITOR	79
3.60 ลักษณะของกระดาษกราฟ	79
3.61 ภาพสำหรับการทดลองปรับค่าพารามิเตอร์ของตัวกรอง BILATERAL	80
3.62 ภาพสำหรับการทดลองหาค่าเทรซโฮลด์	81
3.63 ภาพประกอบการปรับตัวกรองถ่วงน้ำหนัก	81
4.1 โครงสร้างเครื่องเปลี่ยนชุดข้อสอบเทียบกับที่ออกแบบไว้ มุมมองด้านหน้า	83
4.2 โครงสร้างเครื่องเปลี่ยนชุดข้อสอบเทียบกับที่ออกแบบไว้ มุมมองด้านข้าง	83
4.3 โครงสร้างเครื่องเปลี่ยนชุดข้อสอบเทียบกับที่ออกแบบไว้ มุมมองด้านบน	84
4.4 การทดลองที่พบความคลาดเคลื่อน	87
4.5 ชุดข้อสอบไม่เลื่อนออกมา	88
4.6 ชุดข้อสอบเลื่อนออกมาได้ไม่หมดทั้งชุด	89
4.7 ชุดข้อสอบเลื่อนออกมาติดกัน 2 ชุด	89
4.8 ผลจากกระบวนการเทรซโฮลด์ที่ค่าเทรซโฮลด์เท่ากับ 110	93
4.9 ตัวอย่างการเขียนตัวเลขที่ไม่เป็นเชิงเส้นเดียวกัน	94
4.10 การตัดกรอบตัวเลขไม่ครบ	99
4.11 ลักษณะตัวเลขที่เขียนใกล้เคียง	100
4.12 หน้าต่างเริ่มต้น	100
4.13 ผลที่ได้จากการกดปุ่ม “New Answersheet”	101
4.14 ผลที่ได้จากการกดปุ่มเลื่อนตัวกล่อง	102
4.15 ผลที่ได้จากการกดปุ่ม “Detect Box”	103
4.16 ผลที่ได้จากการกดปุ่ม “Ok”	103
4.17 ผลที่ได้จากการใส่ชื่อส่วนที่ลากกรอบตรวจจับละกดปุ่ม “Yes”	104
4.18 ผลที่ได้จากการกดปุ่ม “Ok”	105
4.19 ผลที่ได้จากการกดปุ่ม “Finish”	105
4.20 ผลที่ได้จากการกดปุ่ม “Yes”	106

สารบัญรูป(ต่อ)

รูปที่

4.21 ผลที่ได้จากการกดปุ่ม “Detect”

หน้า

107

สารบัญตาราง

ตารางที่	หน้า	
3.1	กำหนดค่าเริ่มต้นของไมโครคอนโทรลเลอร์	42
3.2	อุปกรณ์อลูมิเนียมโปรไฟล์	68
3.3	อุปกรณ์แผ่นอะคริลิก	68
3.4	อุปกรณ์ฉากอลูมิเนียม และชุดนอตยึดร่อง	69
3.5	อุปกรณ์สายพาน	69
3.6	อุปกรณ์เพลตตรง	69
3.7	อุปกรณ์ท่อตรง	70
3.8	อุปกรณ์แปรงลูกกิ้งตรง	70
3.9	อุปกรณ์ลูกปืนตลับขนาดเล็ก	70
3.10	อุปกรณ์ชุด LEAD SCREW	70
3.11	อุปกรณ์ COUPLING	71
3.12	อุปกรณ์ ARDUINO MEGA 2560	73
3.13	อุปกรณ์ A4988 STEPPER DRIVER	73
3.14	อุปกรณ์ IC L298N	74
3.15	อุปกรณ์ STEPPING MOTOR	74
3.16	อุปกรณ์ DC MOTOR	75
3.17	อุปกรณ์ ULTRASONICS SENSOR MODULE	75
3.18	อุปกรณ์ ENDSTOP SWITCH	75
3.19	คุณสมบัติของคอมพิวเตอร์ ASUS TRANSFORMER BOOK FLIP TP550LD	76
3.20	คุณสมบัติของ WEB CAMERA OKER FULL HD AUTO FOCUS HD- 386	77
4.1	บันทึกกระยะรัศมีที่คลาดเคลื่อนของตำแหน่งที่ป้อนในควอดรันต์ที่ 1	85
4.2	บันทึกกระยะรัศมีที่คลาดเคลื่อนของตำแหน่งที่ป้อนในควอดรันต์ที่ 2	85
4.3	บันทึกกระยะรัศมีที่คลาดเคลื่อนของตำแหน่งที่ป้อนในควอดรันต์ที่ 3	86

สารบัญตาราง (ต่อ)

ตารางที่		หน้า
4.4	บันทึกกระยะรัศมีที่คลาดเคลื่อนของตำแหน่งที่ป้อนในควอดรนต์ที่ 4	86
4.6	ผลการทดลองปรับขนาดของตัวกรอง BILATERAL	90
4.7	ผลการทดลองปรับค่า SIGMA COLOR ของตัวกรอง BILATERAL	90
4.8	ผลการทดลองปรับค่า SIGMA SPACE ของตัวกรอง BILATERAL	91
4.9	ผลการพล็อตกราฟฮิสโตแกรมของพื้นหลังแต่ละสี	92
4.10	ผลการทำงานของตัดกรอบและตัวเลขลาย	94
4.11	ผลการทดลองปรับขนาดของตัวกรองถ่วงน้ำหนัก	95
4.12	ผลการทดลองปรับจำนวนตัวกรองถ่วงน้ำหนักในแต่ละชั้น CONVOLUTION	98
4.13	ผลการทำงานของตัดกรอบและตัวเลขลาย	98

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การรวบรวมคะแนนสอบแบบอัตโนมัติในสถานศึกษาเป็นงานที่ใช้เวลาค่อนข้างมากและมีโอกาสผิดพลาดเกิดขึ้นได้ อีกทั้งยังไม่มีเทคโนโลยีมาช่วยในส่วนนี้ ดังนั้นผู้จัดจึงมีแนวคิดโดยนำเอาเทคโนโลยีด้านคอมพิวเตอร์วิทัศน์ (computer vision) มาช่วยประมวลผล ลดความผิดพลาดของการรวมคะแนนชุดข้อสอบ และมีความรวดเร็วกว่าที่กระทำโดยมนุษย์ เพื่อลดภาระของผู้สอนและสนับสนุนให้มีการวัดผลแบบอัตโนมัติที่เป็นการวัดผลที่มีประสิทธิภาพเมื่อเปรียบเทียบกับข้อสอบแบบปรนัย

ปริญญาานิพนธ์ฉบับนี้ออกแบบระบบรวมคะแนนข้อสอบแบบอัตโนมัติโดยใช้ภาษาไพธอน (Python) ในการพัฒนาโปรแกรม และไมโครคอนโทรลเลอร์ (Microcontroller) สำหรับควบคุมกลไกการเปลี่ยนชุดข้อสอบและการเลื่อนตำแหน่งกล้องที่ใช้จับภาพ โดยกระบวนการทำงานของระบบจะเริ่มจากโปรแกรมรวมคะแนนข้อสอบแบบอัตโนมัติจะรับภาพจากกล้องเพื่อประมวลผลในคอมพิวเตอร์ โดยทำการตัดกรอบและจำแนกตัวเลขโดยใช้ข้อมูลจากกระบวนการเรียนรู้และจดจำ Convolutional neural network (CNN) หลังจากการประมวลผลเสร็จสิ้นจะส่งต่อการทำงานไปยังไมโครคอนโทรลเลอร์ที่ควบคุมตัวป้อนกระดาษ (Feeder and Tray lift) เพื่อเปลี่ยนชุดข้อสอบและยกถาดรองชุดข้อสอบขึ้นให้พอดีกับลูกยางเลื่อนชุดข้อสอบ เมื่อผู้ใช้ต้องการเก็บข้อมูลอื่น ๆ สามารถส่งคำสั่งไมโครคอนโทรลเลอร์ให้ควบคุมชุดเลื่อนตัวกล้องสองแกน (2 Axis Camera) ไปยังตำแหน่งที่ต้องการตรวจจับภาพบนชุดข้อสอบ

1.2 วัตถุประสงค์

- 1) ออกแบบและเขียนโปรแกรมรวมคะแนนข้อสอบจากภาพถ่ายโดยใช้การเรียนรู้เชิงลึก (Deep learning) ร่วมกับการประมวลผลภาพ (Image processing)
- 2) ออกแบบและสร้างเครื่องเปลี่ยนชุดข้อสอบอัตโนมัติ

1.3 ขอบเขตของปริญญาโท

ระบบรวมคะแนนข้อสอบแบบอัตโนมัติแบ่งออกเป็น 3 ส่วน ส่วนแรกคือ ซอฟต์แวร์จำแนกตัวเลขของข้อสอบโดยใช้ภาษาไพธอนร่วมกับ Library Tensorflow ซึ่งเลือกใช้วิธีการแบบ CNN ในการพัฒนาโปรแกรม โปรแกรมจะรับสัญญาณภาพจากกล้องที่ตรวจจับตัวเลขบนชุดข้อสอบได้แล้วนำมาเรียนรู้เชิงลึกเพื่อจำแนกลายมือตัวเลข 0 ถึง 9 ของผู้เขียนได้อย่างแม่นยำ ส่วนที่ 2 คือซอฟต์แวร์ค้นหาตำแหน่งของตัวเลขโดยใช้ภาษาไพธอนร่วมกับ Library OpenCV เพื่อตัดเอาเฉพาะส่วนที่มีตัวเลขก่อนเข้าสู่กระบวนการเรียนรู้เชิงลึกเพื่อจำแนกตัวเลข ผู้ใช้สามารถกำหนดบริเวณการค้นหาตัวเลขจากลายมือได้ เมื่อนำไปใช้กับชุดข้อสอบรูปแบบอื่น ๆ และส่วนที่ 3 เป็นเครื่องเปลี่ยนชุดข้อสอบอัตโนมัติ โดยใช้ชุดไมโครคอนโทรลเลอร์ร่วมกับอุปกรณ์มอเตอร์ขับเคลื่อนสายพาน ที่ควบคุมกลไกเลื่อนชุดข้อสอบ และการเลื่อนตำแหน่งกล้องที่ใช้ในการจับภาพ

บทที่ 2

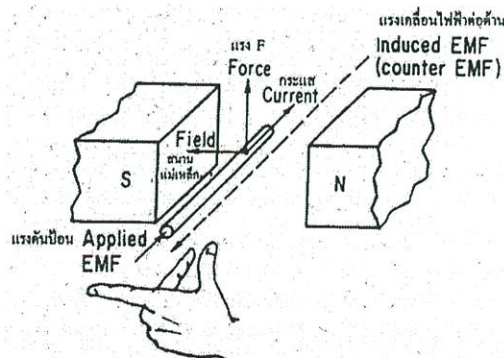
ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 หลักการเกี่ยวกับฮาร์ดแวร์และระบบควบคุม

2.1.1 มอเตอร์ไฟฟ้า

มอเตอร์ไฟฟ้าคืออุปกรณ์ไฟฟ้าที่เปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกล มอเตอร์ไฟฟ้าถูกนำไปใช้งานที่หลากหลายเช่น พัดลมอุตสาหกรรม เครื่องเป่า ปั่น เครื่องมือเครื่องใช้ในครัวเรือน และดิสก์ไดรฟ์ เป็นต้น มอเตอร์ไฟฟ้าสามารถขับเคลื่อนโดยแหล่งจ่ายไฟกระแสตรง (DC) เช่น จากแบตเตอรี่ ยานยนต์ และวงจรเรียงกระแส เป็นต้น และจากแหล่งจ่ายไฟกระแสสลับ (AC) เช่น จากบ้าน อินเวอร์เตอร์ และเครื่องปั่นไฟ เป็นต้น

หลักการทำงานของมอเตอร์ไฟฟ้าคือ เมื่อมีกระแสไฟฟ้าเข้าไปยังขดลวดในสนามแม่เหล็กจะทำให้เกิดแรงแม่เหล็กซึ่งมีสัดส่วนของแรงขึ้นกับกระแสแรงของสนามแม่เหล็ก โดยแรงจะเกิดขึ้นเป็นมุมฉากกับกระแสและสนามแม่เหล็ก ขณะที่ทิศทางของแรงกลับตรงกันข้ามกัน ถ้าหากกระแสของสนามแม่เหล็กไหลย้อนกลับจะทำให้เกิดการเปลี่ยนแปลงของกระแส และสนามแม่เหล็กเป็นผลทำให้ทิศทางของแรงเปลี่ยนไป ด้วยคุณสมบัตินี้ทำให้มอเตอร์กระแสตรงกลับทิศทางการทำงานได้ เนื่องจากมีความสัมพันธ์ระหว่างทิศทางของสนามแม่เหล็ก ทิศทางของกระแสไฟฟ้าในตัวนำ และทิศทางที่ตัวนำเคลื่อนที่ ซึ่งความสัมพันธ์เหล่านี้สามารถตั้งเป็นกฎมอเตอร์แสดงดังรูปที่ 2.1 โดยกำหนดให้หัวแม่มือ นิ้วชี้ และนิ้วกลาง ให้ตั้งฉากซึ่งกันและกัน โดยหัวแม่มือชี้ตามทิศทางของแรงเคลื่อนที่ของตัวนำ (Force) นิ้วชี้ชี้ตามทิศทางของสนามแม่เหล็ก (Field) และนิ้วกลางชี้ไปตามทิศทางการไหลของกระแสไฟฟ้า (Current)



รูปที่ 2.1 กฎมอเตอร์ [1]

2.1.1.1 มอเตอร์กระแสตรง (DC Motor)

มอเตอร์กระแสตรง (DC Motor) คือเครื่องกลที่เปลี่ยนพลังงานไฟฟ้าไปเป็นพลังงานกล โดยอาศัยหลักการคือเมื่อกระแสไหลผ่านตัวนำซึ่งวางอยู่ในสนามแม่เหล็กจะมีแรงเกิดขึ้นที่ลวดตัวนำ ทำให้ลวดตัวนำเกิดการเคลื่อนที่ซึ่งเกิดขึ้นจากแรงแม่เหล็ก โดยส่วนของแรงนี้จะขึ้นอยู่กับกระแสและกำลังของสนามแม่เหล็ก นิยมใช้กันอย่างแพร่หลายในโรงงานต่างๆ เป็นอุปกรณ์ที่ใช้ควบคุมเครื่องจักรกลต่างๆในงานอุตสาหกรรม สามารถนำมาประยุกต์ใช้กับการคอนโทรลมอเตอร์กระแสตรงได้ จึงทำให้ได้รับความนิยมอย่างมากในปัจจุบัน

มอเตอร์กระแสตรงมีผลการทำงานที่ไม่มีตำแหน่งหยุดที่แน่นอน ดังนั้นระบบคอนโทรลมอเตอร์กระแสตรงจึงเป็นแบบวงรอบปิด ตำแหน่งเอาต์พุตหรือความเร็วเอาต์พุตจะถูกป้อนกลับไปเปรียบเทียบกับอินพุตอ้างอิงเพื่อให้ได้ผลการทำงานตามที่ต้องการ

มอเตอร์ไฟฟ้ากระแสตรงมีโครงสร้างและส่วนประกอบเหมือนกันกับเครื่องกำเนิดไฟฟ้ากระแสตรงสามารถแบ่งชนิดได้เป็น มอเตอร์ไฟฟ้ากระแสตรงแบบแยกวงจรถะตุ้นสนามแม่เหล็ก (Separately excited DC Motor) มอเตอร์ไฟฟ้ากระแสตรงแบบกระตุ้นในตัวเอง (Self excited DC Motor) มอเตอร์ไฟฟ้ากระแสตรงแบบซีรีย์ (DC series Motor) มอเตอร์ไฟฟ้ากระแสตรงแบบชันท (DC shunt Motor) มอเตอร์ไฟฟ้ากระแสตรงแบบคอมเปานด์ (DC compound Motor) ตัวอย่างของอุปกรณ์มอเตอร์กระแสตรงแสดงได้ดังรูปที่ 2.2



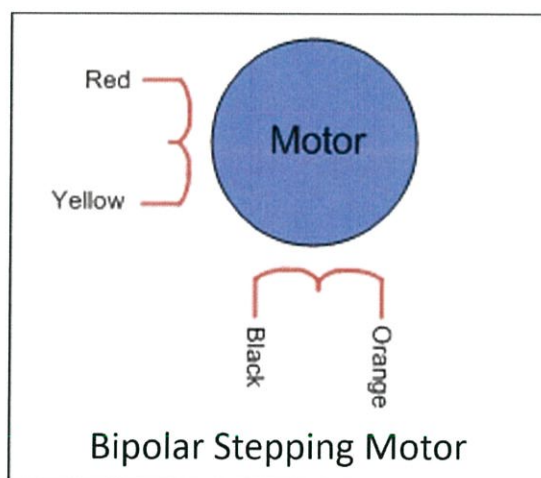
รูปที่ 2.2 มอเตอร์กระแสตรง [2]

2.1.1.2 สเต็ป핑มอเตอร์ (Stepping Motor)

สเต็ป핑มอเตอร์ (Stepping Motor) เป็นอุปกรณ์เอาต์พุตที่สามารถนำไอซี และไมโครคอนโทรลเลอร์มาทำการควบคุมได้ง่าย และเป็นมอเตอร์ที่เหมาะสมกับการใช้ในงานควบคุม

การหมุนที่ต้องการตำแหน่งและทิศทางที่แน่นอน การทำงานของ Stepping Motor จะขับเคลื่อนด้วยพัลส์ ลักษณะการขับเคลื่อนจะหมุนรอบแกนได้ 360 องศา มีลักษณะไม่ต่อเนื่องแต่มีลักษณะเป็นขั้นเคลื่อนก้าวละ (Step) 0.9, 1.8, 5, 7.5, 15 หรือ 50 องศา ซึ่งขึ้นอยู่กับคุณสมบัติแต่ละชนิดของ Stepping Motor แต่ละตัว การควบคุมและสั่งงานให้ Stepping Motor ทำงานไปที่ละก้าวสามารถทำได้โดยการจ่ายกำลังไฟไปยังขดลวดในแต่ละขอบของสเตเตอร์ โดยการป้อนจะทำให้ในลักษณะเป็นลำดับ การกำหนดชนิด Stepping Motor พิจารณาที่ลักษณะของการพันขดลวด, การต่อสายออกมาใช้งาน และวงจรขับ ซึ่งแบ่งเป็น 2 ชนิดคือ ชนิดไบโพลาร์ (bipolar) และชนิดยูนิโพลาร์ (unipolar)

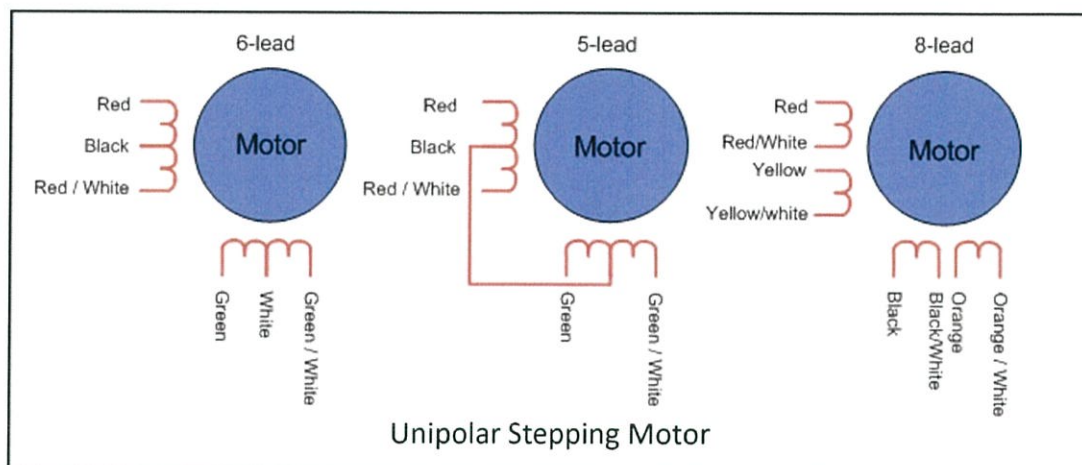
Stepping Motor แบบไบโพลาร์มีลักษณะการพันขดลวดของมอเตอร์แสดงดังรูปที่ 2.3 แบ่งออกเป็น 2 ขดที่ไม่มีแท่งกลาง ทำให้บางครั้งจึงเรียก Stepping Motor แบบนี้ว่า Stepping Motor แบบ 2 เฟส การขับให้มอเตอร์แบบนี้หมุนจะต้องป้อนแรงดันต่างขั้วกันให้แก่ขดลวดแต่ละขด ทำให้วงจรขับ Stepping Motor แบบนี้ค่อนข้างซับซ้อน



รูปที่ 2.3 ลักษณะการพันขดลวดของ Stepping Motor แบบไบโพลาร์ [3]

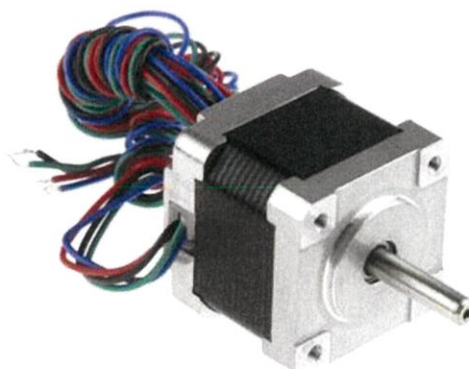
Stepping Motor แบบยูนิโพลาร์มีลักษณะการพันขดลวดของมอเตอร์แสดงในรูปที่ 2.4 มีด้วยกัน 3 แบบคือ แบบ 6 สาย 5 สาย และ 8 สาย บางครั้งเรียก Stepping Motor แบบนี้ว่าเป็น Stepping Motor แบบ 4 เฟส การขับจะต้องป้อนสัญญาณเข้าที่ขั้วหรือเฟสของมอเตอร์ให้เรียงลำดับอย่างถูกต้อง มอเตอร์จึงจะสามารถหมุนได้อย่างราบรื่น Stepping Motor แบบนี้มีการพันขดลวด 2 ขดบนแต่ละขั้วแม่เหล็กของสเตเตอร์ แต่ละขดแบ่งเป็น 2 เฟส รวมมอเตอร์ทั้งตัวจะมี 4 เฟสคือ เฟส 1, 2, 3 และ 4 มีการต่อสายออกมาจากขดลวดแต่ละขดเพื่อจ่ายไฟเลี้ยง ทำ

Stepping Motor แบบนี้มีทั้งแบบ 5 สายและ 6 สาย ถ้าเป็นแบบ 5 สาย จะเป็นการนำสายไฟเลี้ยงของขดลวดทั้งสองมาต่อรวมกันเป็นสายเดียว สำหรับในบทความนี้จะเน้นหนักไปที่สเต็ปเปอร์แบบยูนิโพลาร์นี้ เนื่องจากสามารถหาได้ง่ายกว่า และใช้วงจรขับที่มีความซับซ้อนน้อยกว่ามาก



รูปที่ 2.4 ลักษณะการพันขดลวดของ Stepping Motor แบบยูนิโพลาร์ [3]

โดยส่วนใหญ่จะใช้ Stepping Motor มาทำการการควบคุมโดยใช้วิธีในระบบดิจิทัล เช่น พรินเตอร์ (Printer) พล็อตเตอร์ (X-Y Plotter) ดิสก์ไดรฟ์ (Disk drive) เป็นต้น Stepping Motor แสดงได้ดังรูปที่ 2.5

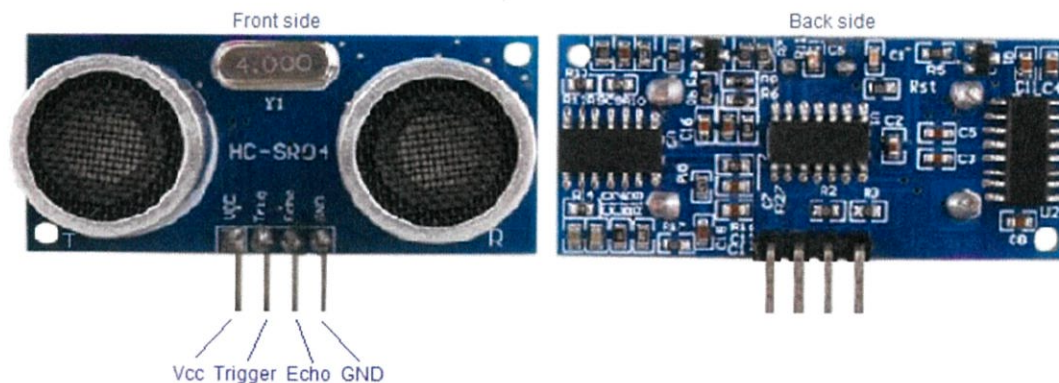


รูปที่ 2.5 Stepping Motor [4]

2.1.2 Ultrasonic Sensor module

Ultrasonic Sensor module หรือ HC-SR04 นี้เป็นอุปกรณ์ใช้วัดระยะทางโดยไม่ต้องมีการสัมผัสกับตำแหน่งที่ต้องการวัด วัดได้ตั้งแต่ 2 cm ถึง 400 cm โดยส่งสัญญาณอัลตราโซ

นิกความถี่ 40 kHz ไปในอากาศ ถ้าพบวัตถุ หรือสิ่งกีดขวางบนเส้นทาง คลื่น Ultrasound นั้นจะ ถูกสะท้อนกลับไปยัง Ultrasonic Sensor module แสดงดังรูปที่ 2.7 ซึ่งสามารถคำนวณหาระยะ ได้จากความเร็วของเสียงและระยะเวลาที่เสียงใช้การเดินทาง ตัวอย่างของ Ultrasonic Sensor module แสดงได้ดังรูปที่ 2.6



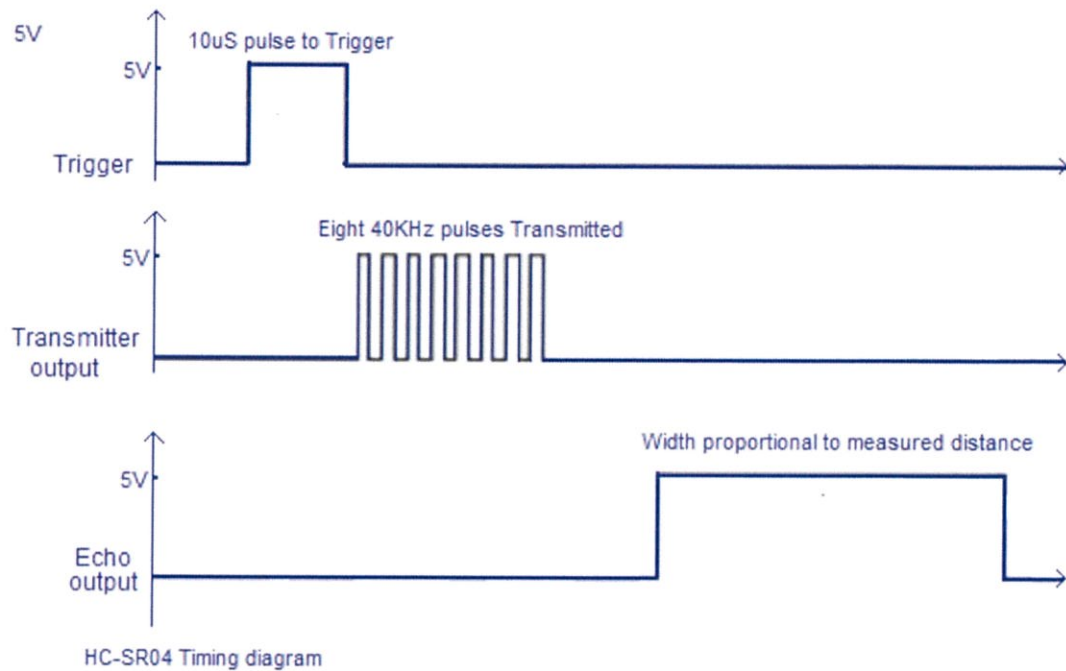
รูปที่ 2.6 Ultrasonic Sensor module [5]



รูปที่ 2.7 หลักการทำงานของ Ultrasonic Sensor module [5]

จากรูปที่ 2.6 Ultrasonic Sensor module มีพินเชื่อมต่อทั้งหมด 4 พินคือ GND VCC Trig Echo และ GND ในการสร้างคลื่น Ultrasound ต้องตั้งค่าสถานะของพิน Trig เป็น High เป็นเวลา 10 μ s โดยจะส่งคลื่น Ultrasound เป็นรอบ รอบละ 8 จังหวะที่เดินทางด้วยความเร็ว

เสียง และจะรับสัญญาณสะท้อนกลับที่พิน Echo แล้วส่งข้อมูลเวลาที่ใช้ในการเดินทางเป็น Output แสดงดังรูปที่ 2.8



รูปที่ 2.8 การส่ง และรับสัญญาณของ Ultrasonic Sensor module [5]

ตัวอย่างในการคำนวณ โดยกำหนดให้วัตถุอยู่ห่างจาก Ultrasonic Sensor module 10 cm และความเร็วของเสียงคือ 340 m/s หรือ 0.034 cm/µs ซึ่งสามารถประมาณค่าเวลาที่คลื่น Ultrasound ใช้ในการเดินทางได้จากการนำระยะห่างของวัตถุกับ Ultrasonic Sensor module มาหารด้วยความเร็วของเสียง จึงสามารถเขียนเป็นสมการได้ดังสมการที่ 2.1

$$t = \frac{s}{v} \quad (2.1)$$

- เมื่อ t แทนเวลาที่คลื่น Ultrasound ใช้ในการเดินทาง
 s แทนระยะทาง
 v แทนความเร็วของเสียงคือ 340 m/s หรือ 0.034 cm/µs

จากสมการที่ 2.1 แต่พิน Echo จะส่งค่าเวลาที่คลื่น Ultrasound ใช้ในการเดินทาง ประมาณ 294 µs แต่ค่าที่ได้มีค่าเป็นสองเท่าของระยะทางจริง เนื่องจากคลื่น Ultrasound เดินทาง

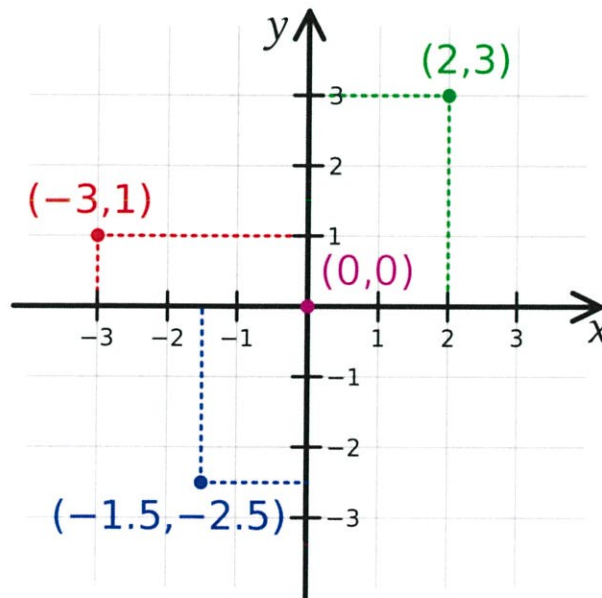
ไปข้างหน้าและสะท้อนกลับมา ดังนั้นเพื่อให้ได้ระยะทางที่เป็นหน่วยเซนติเมตร จึงต้องนำค่าเวลาที่
ได้รับจากพิน Echo คูณด้วยความเร็วเสียง 0.034 และหารด้วย 2 แสดงดังสมการที่ 2.2

$$s = \frac{1}{2} \times \frac{t}{v} \quad (2.2)$$

ดังนั้นการใช้งาน และการเขียนโปรแกรมควบคุม Ultrasonic Sensor module
เพื่อให้ได้ค่าระยะทางที่ถูกต้อง จึงต้องนำค่าที่รับได้จากพิน Echo มาคำนวณในสมการที่ 2.2 ก่อน

2.1.3 ทฤษฎี CoreXY

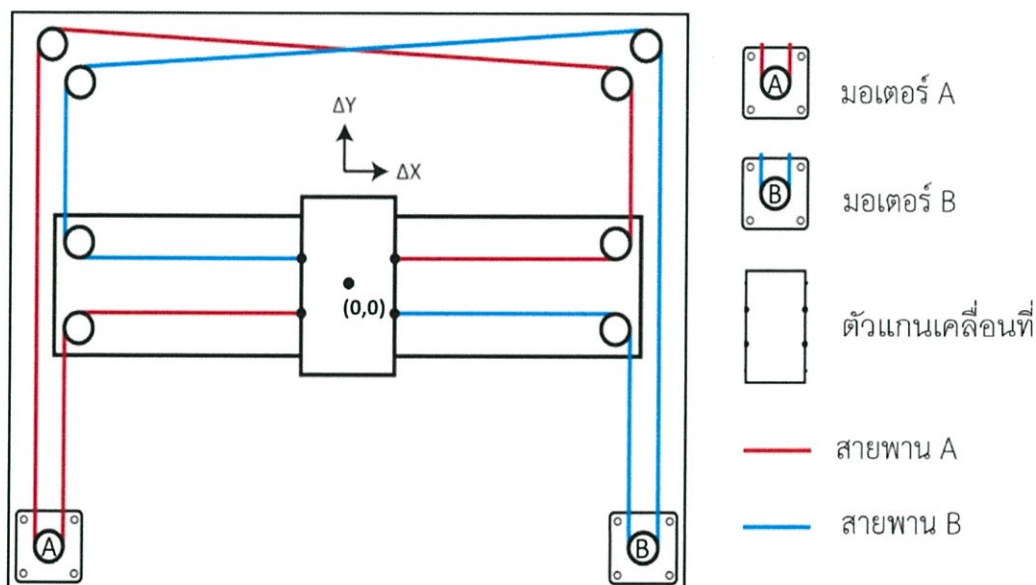
ทฤษฎี CoreXY เป็นระบบการเคลื่อนที่ที่เป็นองค์ประกอบที่สำคัญของเครื่องพิมพ์สามมิติ
โดยเคลื่อนที่ในระบบพิกัดคาร์ทีเซียน ที่เป็นระบบที่ใช้กำหนดตำแหน่งของจุดแต่ละจุดบนระนาบ โดย
อ้างอิงตัวเลข 2 จำนวน ซึ่งแต่ละจำนวนเรียกว่า พิกัด X และพิกัด Y ของจุดนั้น และเพื่อกำหนดพิกัด
ของจุดจะต้องมีเส้นแกนสองเส้นตัดกันเป็นมุมฉากที่จุดกำเนิดได้แก่ แกน X และแกน Y ซึ่งเส้นแกน
ดังกล่าวจะมีหน่วยบอกความยาวเป็นระยะ ระบบพิกัดคาร์ทีเซียนยังสามารถใช้ในปริภูมิสามมิติ ซึ่ง
จะมีแกน Z และพิกัด Z เพิ่มเข้ามา สามารถอธิบายดังรูปที่ 2.9 โดยทฤษฎี CoreXY เป็นการเคลื่อนที่ใน
เฉพาะในระบบพิกัดคาร์ทีเซียน 2 แกน คือแกน X และแกน Y



รูปที่ 2.9 ระบบพิกัดคาร์ทีเซียน [6]

จากรูปที่ 2.9 ระบบพิกัดคาร์ทีเซียนที่จุด (2,3) สีเขียวมีพิกัด X เท่ากับ 2 และพิกัด Y เท่ากับ 3 จุด (-3,1) สีแดงมีพิกัด X เท่ากับ -3 และพิกัด Y เท่ากับ 1 จุด (-1.5,-2.5) สีน้ำเงินมีพิกัด X เท่ากับ -1.5 และพิกัด Y เท่ากับ -2.5 และจุด (0,0) สีม่วงซึ่งเป็นจุดกำเนิด

การทำงานของ CoreXY ต้องใช้ Stepping Motor 2 ตัว คือมอเตอร์ A และมอเตอร์ B ในการควบคุมและใช้สายพาน 2 เส้น คือสายพาน A และสายพาน B โดยสายพานทั้งสองจะยึดกับตัวแกนเคลื่อนที่เพื่อรับแรงกระทำจาก Stepping Motor และการจัดวางตำแหน่งของ Stepping Motor และสายพานเป็นดังรูปที่ 2.10



รูปที่ 2.10 การจัดวางตำแหน่งของ Stepping Motor และสายพาน [7]

ระยะที่ต้องการในการเลื่อนตัวแกนเคลื่อนที่จะถูกควบคุมโดยระยะการหมุน และทิศทางการหมุนของ Stepping Motor ทั้งสองตัว โดยแทนระยะที่ต้องการเลื่อนตัวกอลังในแนวแกน X เป็น ΔX แทนระยะที่ต้องการเลื่อนตัวกอลังในแนวแกน Y เป็น ΔY แทนระยะที่มอเตอร์ A หมุนในทิศทางทวนเข็มนาฬิกาเป็น ΔA และแทนระยะที่มอเตอร์ B หมุนในทิศทางทวนเข็มนาฬิกาเป็น ΔB โดยมีความสัมพันธ์ดังสมการที่ 2.1 และสมการที่ 2.2 และความสัมพันธ์ระหว่างทิศทางการหมุนกับตัวแปร ΔA และ ΔB แสดงดังรูปที่ 2.11

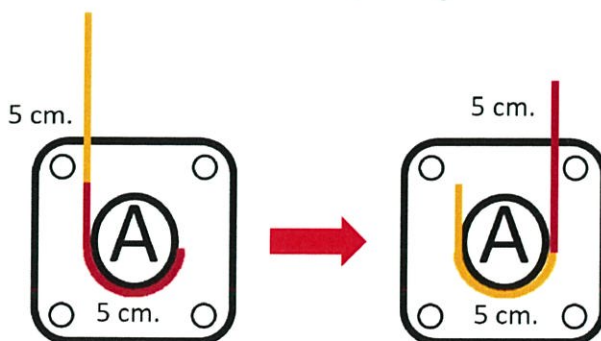
$$\Delta X = \frac{1}{2}(\Delta A + \Delta B) \quad (2.3)$$

$$\Delta Y = \frac{1}{2}(\Delta A - \Delta B) \quad (2.4)$$

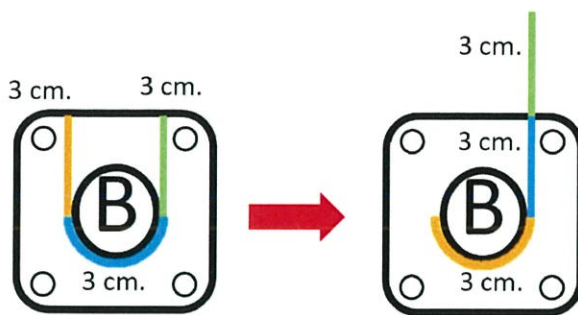


รูปที่ 2.11 ความสัมพันธ์ระหว่างทิศทางการหมุนกับตัวแปร ΔA และ ΔB

จากรูปที่ 2.9 หากต้องการให้ตัวแกนเคลื่อนที่เคลื่อนที่ไปยังพิกัด (2,3) เซนติเมตร จากจุดกำเนิด (0,0) เซนติเมตร และต้องการทราบค่าระยะที่มอเตอร์ A ต้องหมุนแทนด้วย ΔA และระยะที่มอเตอร์ B ต้องหมุนแทนด้วย ΔB โดยขั้นแรกหาค่า ΔX จากการนำค่าพิกัดในแนวแกน X ที่จุดปลายทางลบด้วยที่จุดเริ่มต้น นั่นคือ $2-0=2$ เซนติเมตร และหาค่า ΔY จากการนำค่าพิกัดในแนวแกน Y ที่จุดปลายทางลบด้วยที่จุดเริ่มต้น นั่นคือ $3-0=3$ เซนติเมตร จากนั้นนำ $\Delta X=2$ เซนติเมตร และ $\Delta Y=3$ เซนติเมตร แทนลงในสมการที่ 2.3 และสมการที่ 2.4 จึงทราบค่า $\Delta A=5$ เซนติเมตร คือมอเตอร์ A ต้องหมุนทิศทางทวนเข็มนาฬิกาไป 5 เซนติเมตร และ $\Delta B=-1$ คือมอเตอร์ B ต้องหมุนทิศทางตามเข็มนาฬิกาไป 1 เซนติเมตร โดยสามารถอธิบายระยะที่มอเตอร์ A หมุนได้ดังรูปที่ 2.12 และอธิบายระยะที่มอเตอร์ B หมุนได้ดังรูปที่ 2.13



รูปที่ 2.12 ระยะที่มอเตอร์ A หมุนได้

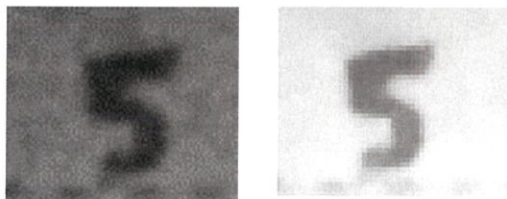


รูปที่ 2.13 ระยะที่มอเตอร์ B หมุนได้

2.2 หลักการเกี่ยวกับการประมวลผลภาพ

2.2.1 ตัวกรอง Bilateral (Bilateral Filter)

ตัวกรอง Bilateral เป็นตัวกรองไม่เป็นเชิงเส้น (nonlinear) โดยมีลักษณะพิเศษที่สามารถลดสัญญาณรบกวนได้ และยังคงรักษาความคมชัดของขอบในภาพไปพร้อมกันด้วยสามารถแสดงได้ดังรูปที่ 2.14



รูปที่ 2.14 การเปรียบเทียบภาพต้นฉบับกับภาพที่ผ่านตัวกรอง Bilateral

จากรูปที่ 2.14 จะพบว่าขอบของรูปร่างวัตถุภายในภาพจะชัดเจนมากขึ้น และรายละเอียดของภาพจะลดลง เนื่องจากตัวกรองจะแทนค่าของแต่ละพิกเซลด้วยค่าเฉลี่ยถ่วงน้ำหนักของพิกเซลใกล้เคียง โดยค่าถ่วงน้ำหนักสามารถอิงตามการกระจายแบบเกาส์เซียน ความพิเศษตัวของตัวกรองนี้คือค่าถ่วงน้ำหนักไม่ได้ขึ้นอยู่กับพิกเซลใกล้เคียงที่คำนวณจากระยะทางแบบยุคลิดเท่านั้น แต่ขึ้นอยู่กับความแตกต่างระหว่างค่าของแต่ละพิกเซลด้วย ซึ่งทำให้รักษาขอบของวัตถุในภาพให้มีความคมชัดได้ โดยตัวกรอง Bilateral สามารถนิยามได้ดังสมการที่ 2.5

$$I^b(x) = \frac{1}{W_p} \sum_{x_i \in \Omega} f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|) \quad (2.5)$$

โดยที่ W_p สามารถคำนวณได้จากสมการที่ 2.6

$$W_p = \sum_{x_i \in \Omega} f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|) \quad (2.6)$$

- เมื่อ I^b แทนภาพที่ถูกกรอง
 I แทนภาพอินพุตของตัวกรอง
 x แทนพิกัดของพิกเซลปัจจุบันที่ถูกกรอง
 Ω แทนกรอบของพิกเซลใกล้เคียงที่มีพิกเซลปัจจุบันเป็นตำแหน่งกลาง
 f_r แทนฟังก์ชันหาเคอร์เนลช่วงที่ทำให้เกิดความแตกต่างของความเข้ม
 g_s แทนฟังก์ชันหาเคอร์เนลเชิงพื้นที่เพื่อให้เกิดความแตกต่างในพิกัด

ฟังก์ชันหาเคอร์เนลช่วงที่ทำให้เกิดความแตกต่างของความเข้ม และฟังก์ชันหาเคอร์เนลเชิงพื้นที่เพื่อให้เกิดความแตกต่างในพิกัดสามารถใช้เป็นฟังก์ชันการกระจายแบบเกาส์เซียนได้

2.2.2 เทรชโฮลด์ (Thresholding) และฮิสโตแกรม (Histogram)

เทรชโฮลด์เป็นกระบวนการอย่างง่ายในการแบ่งสัดส่วนของภาพออกเป็นสองส่วนคือขาวและดำ ดังนั้นเมื่อผ่านกระบวนการเทรชโฮลด์ภาพขาวดำหรือภาพสีจะถูกเปลี่ยนเป็นภาพภาพไบนารี วิธีการทำได้โดยกำหนดค่าเทรชโฮลด์เพื่อใช้ในเปรียบเทียบกับค่าแต่ละพิกเซลหากความเข้มของพิกเซลสูงกว่าเทรชโฮลด์ พิกเซลจะถูกกำหนดให้เป็นสีขาวในเอาต์พุต หากความเข้มของพิกเซลต่ำกว่าเทรชโฮลด์จะกำหนดเป็นสีดำ สามารถเขียนได้ดังสมการที่ 2.7

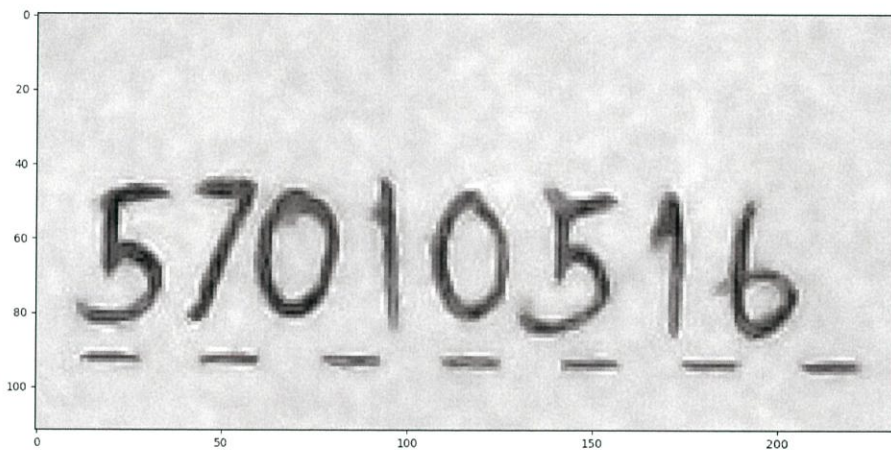
$$I^T(x, y) = \begin{cases} 1, & I(x, y) \geq T \\ 0, & I(x, y) < T \end{cases} \quad (2.7)$$

- เมื่อ I^T แทนความเข้มพิกเซลที่ผ่านกระบวนการเทรชโฮลด์
 I แทนความเข้มพิกเซลอินพุต
 x แทนพิกัดของพิกเซลตามแนวแกนนอน
 y แทนพิกัดของพิกเซลตามแนวแกนตั้ง
 T แทนค่าเทรชโฮลด์

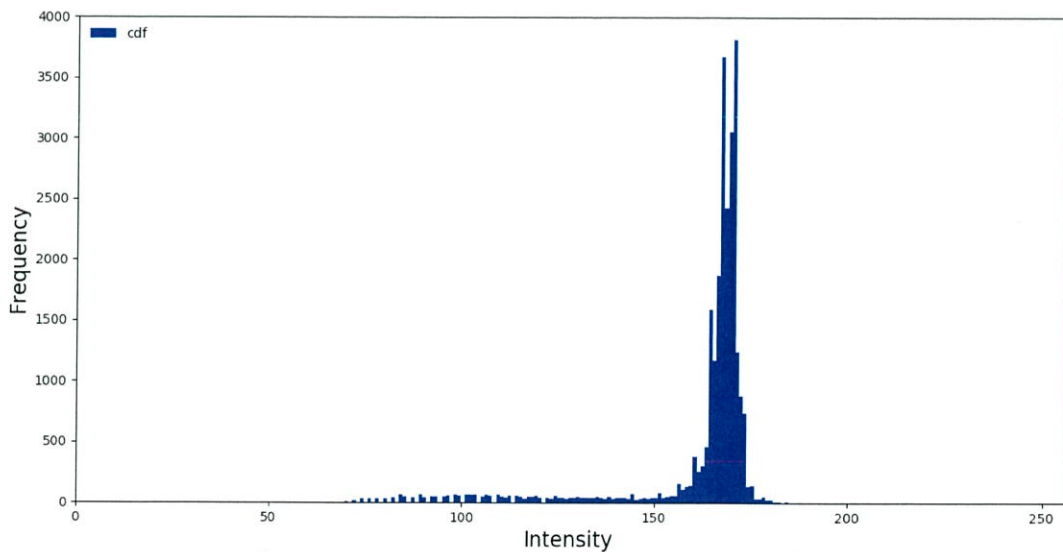
จากสมการที่ 2.7 การกำหนดให้ค่าความเข้มพิกเซลเป็น 0 จะแทนสีดำและกำหนดค่าความเข้มพิกเซลเป็น 1 จะแทนสีขาว ซึ่งสามารถกำหนดค่าความเข้มพิกเซลสลับกันได้ขึ้นอยู่กับการใช้งานหลังจากผ่านกระบวนการนี้ ซึ่งเทรชโฮลด์เป็นหนึ่งในวิธีการแยกวัตถุออกจากพื้นหลังในภาพ

เทอร์ชโฮลด์ต้องอาศัยการกำหนดค่าเทอร์ชโฮลด์ ดังนั้นการพิจารณาเลือกค่าเทอร์ชโฮลด์ที่เหมาะสมต้องใช้ในการพล็อตกราฟฮิสโตแกรม

กราฟฮิสโตแกรมที่แสดงความสัมพันธ์ระหว่างหมวดหมู่ของข้อมูลหรือชั้นของข้อมูลกับความถี่ของข้อมูลนั้น ตัวอย่างเช่นภาพขาวดำที่เก็บความเข้มของพิกเซลด้วยบิต 8 บิตจะพบว่ามีช่วงของข้อมูลอยู่ที่ 0 ถึง 255 ซึ่งเมื่อนำมาพล็อตกราฟฮิสโตแกรมจะได้แกนนอนเป็นค่าความเข้มพิกเซลและแกนตั้งเป็นความถี่ของความเข้มนั้น ตัวอย่างเช่น เมื่อนำภาพที่รูปที่ 2.15 ไปพล็อตกราฟฮิสโตแกรมแสดงได้ดังรูปที่ 2.16

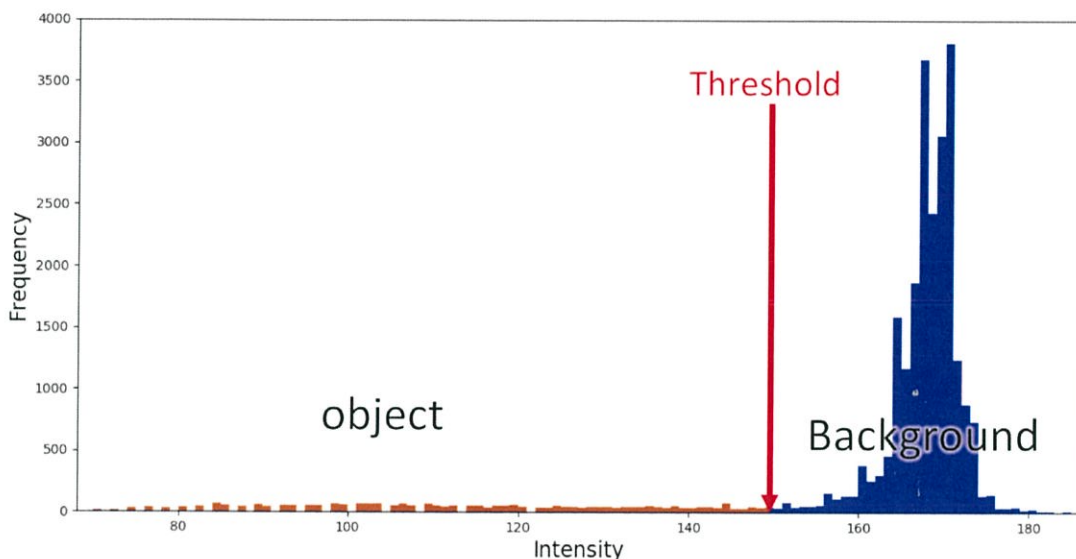


รูปที่ 2.15 ภาพตัวเลขลายมือ



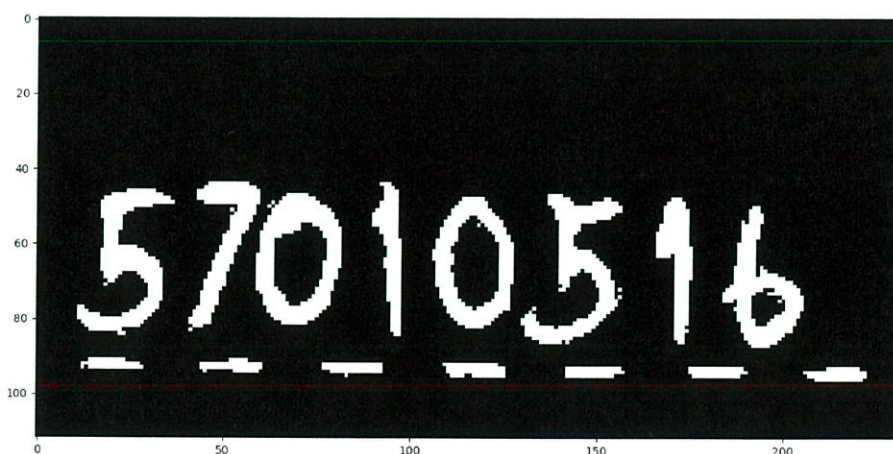
รูปที่ 2.16 กราฟฮิสโตแกรมความเข้มพิกเซลของรูปที่ 2.15

จากรูปที่ 2.15 และ 2.16 จะพบว่าสามารถพิจารณาเลือกค่าเทรชโฮลด์ได้จากกราฟฮิสโตแกรม เพราะความเข้มของพิกเซลที่เป็นพื้นหลังจะมีค่าใกล้เคียงกันและมีจำนวนมากหลายพิกเซล ทำให้เกิดความถี่มากในระดับความเข้มที่เป็นพื้น และส่วนวัตถุจะมีการกระจายตัวเป็นอีกกลุ่มในระดับความเข้มที่ต่ำกว่า และเมื่อกำหนดให้เทรชโฮลด์เท่ากับ 150 จะสามารถแบ่งส่วนของภาพได้เป็น 2 ส่วนดังรูปที่ 2.17



รูปที่ 2.17 การกำหนดค่าเทรชโฮลด์ทำให้เกิดการแบ่งส่วนของภาพ

จากการกำหนดค่าให้เทรชโฮลด์เท่ากับ 150 และนำรูปที่ 2.15 เข้าสู่กระบวนการเทรชโฮลด์จะได้ภาพดังรูปที่ 2.18



รูปที่ 2.18 ผลจากการนำรูปที่ 2.16 เข้าสู่กระบวนการเทรชโฮลด์

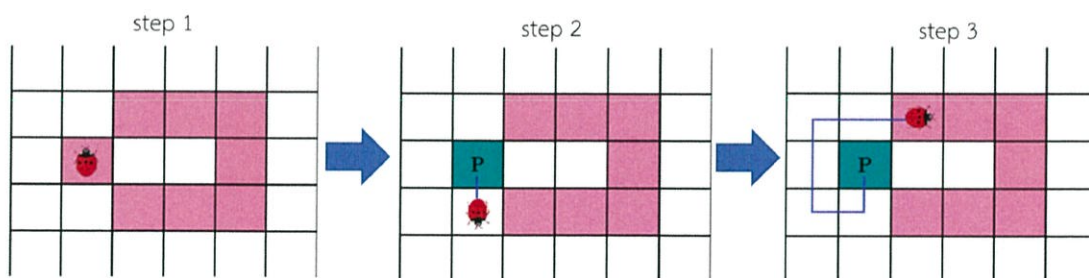
2.2.3 การหาคอนทัวร์ (Find Contours)

คอนทัวร์สามารถอธิบายได้ว่าเป็นเส้นโค้งที่เชื่อมต่อกับจุดที่ต่อเนื่องกันทั้งหมด (ตามแนวเขต) ที่มีสีเดียวกันหรือความเข้มเดียวกันคอนทัวร์ เป็นเครื่องมือที่มีประโยชน์สำหรับการวิเคราะห์รูปร่าง การตรวจจับวัตถุและการจดจำ

การหาคอนทัวร์ในภาพสามารถทำได้หลากหลายวิธี ยกตัวอย่างวิธี Moor Neighbor Tracing โดยมีหลักการดังนี้

- 1) หาดำแหน่งพิกเซลเริ่มต้นด้วยการสแกนจากตำแหน่งมุมล่างซ้าย โดยเลื่อนจากล่างขึ้นบน และจากซ้ายไปขวา ซึ่งให้กำหนดภาพเป็นภาพไบนารี
- 2) เมื่อพบพิกเซลที่มีสีขาว โปรแกรมจะเก็บค่าตำแหน่งค่าพิกัดและย้อนกลับไป 1 ตำแหน่งก่อนพบพิกเซลสีขาว
- 3) เริ่มการสแกนด้วยรูปแบบตามเข็มนาฬิกาไปจนกว่าจะพบพิกเซลสีขาว
- 4) ทำขั้นตอนที่ 2 ถึง 3 ซ้ำจนกว่าตำแหน่งที่เก็บได้จะซ้ำกับตำแหน่งแรกจึงสิ้นสุดการหาหาคอนทัวร์

โดยกระบวนการหาคอนทัวร์ด้วยวิธี Moor Neighbor Tracing สามารถอธิบายได้ดังรูปที่ 2.19 โดยกำหนดให้สีชมพูแทนพิกเซลสีขาว สีเทาแทนพิกเซลสีดำ และสีเขียวแทนพิกเซลที่ถูกบันทึกตำแหน่ง



รูปที่ 2.19 กระบวนการหาคอนทัวร์ด้วยวิธี Moor Neighbor Tracing

2.3.4 การเปรียบเทียบภาพอ้างอิง (Template Matching)

หลักการเปรียบเทียบภาพอ้างอิงจะเปรียบเทียบพิกเซลของภาพว่ามีความเหมือนกันหรือไม่ การเปรียบเทียบภาพอ้างอิงเป็นวิธีที่ใช้สำหรับการค้นหาและตรวจสอบตำแหน่งของ

ภาพเทมเพลตซึ่งมีขนาดเล็กกว่าภาพที่นำมาเปรียบเทียบ ซึ่งการเปรียบเทียบสามารถใช้ความเข้มของพิกเซลโดยใช้การวัด SAD (Sum of absolute differences) ซึ่งความแตกต่างสัมบูรณ์ของความเข้มพิกเซลสามารถแสดงได้ดังสมการที่ 2.8

$$Diff(x_s, y_s, x_t, y_t) = |I_s(x_s, y_s) - I_t(x_t, y_t)| \quad (2.8)$$

เมื่อ	$Diff$	แทนฟังก์ชันการหาความแตกต่างสัมบูรณ์ของความเข้มพิกเซล
	I_s	แทนความเข้มพิกเซลภาพอินพุต
	I_t	แทนความเข้มพิกเซลภาพอ้างอิง (Template)
	x_s, y_s	แทนพิกัดพิกเซลของภาพอินพุต
	x_t, y_t	แทนพิกัดพิกเซลของภาพอ้างอิง (Template)

จากสมการที่ 2.8 สามารถเขียนเป็นฟังก์ชันการหาผลรวมของความแตกต่างสัมบูรณ์ (Sum of absolute differences) ได้ดังสมการที่ 2.9

$$SAD(x, y) = \sum_{i=0}^{T_{rows}} \sum_{j=0}^{T_{cols}} Diff(x+i, y+j, i, j) \quad (2.9)$$

เมื่อ	SAD	แทนฟังก์ชันการหาผลรวมของความแตกต่างสัมบูรณ์
	x, y	แทนพิกัดพิกเซลของภาพอินพุต
	T_{rows}	แทนจำนวนแถวทั้งหมดของภาพอ้างอิง (ขนาดความยาว)
	T_{cols}	แทนจำนวนหลักทั้งหมดของภาพอ้างอิง (ขนาดความกว้าง)

จากสมการที่ 2.9 สามารถเขียนใช้กระบวนการทางคณิตศาสตร์เพื่อสแกนหาภาพอ้างอิงในภาพอินพุตได้โดยการเลื่อนพิกัดเริ่มต้นในฟังก์ชันการหาผลรวมของความแตกต่างสัมบูรณ์ไปจนครบทั้งภาพอินพุตแสดงได้ดังสมการที่ 2.10

$$M(x, y) = \sum_{i=0}^{S_{rows}} \sum_{j=0}^{S_{cols}} SAD(x, y) \quad 2.10$$

เมื่อ	M	แทนฟังก์ชันผลการเปรียบเทียบผลต่าง
-------	-----	-----------------------------------

x, y	แทนพิกัดพิกเซลของภาพอินพุต
S_{rows}	แทนจำนวนแถวทั้งหมดของภาพอินพุต (ขนาดความยาว)
S_{cols}	แทนจำนวนหลักทั้งหมดของภาพอินพุต(ขนาดความกว้าง)

จากสมการที่ 2.10 หากค่าจากฟังก์ชันผลการเปรียบเทียบผลต่างในพิกัดใด มีค่าน้อย แสดงว่ามีโอกาสที่ตำแหน่งนั้นพบภาพที่เหมือนกับภาพอ้างอิง

2.4 หลักการที่ใช้ในกระบวนการเรียนรู้และจดจำ

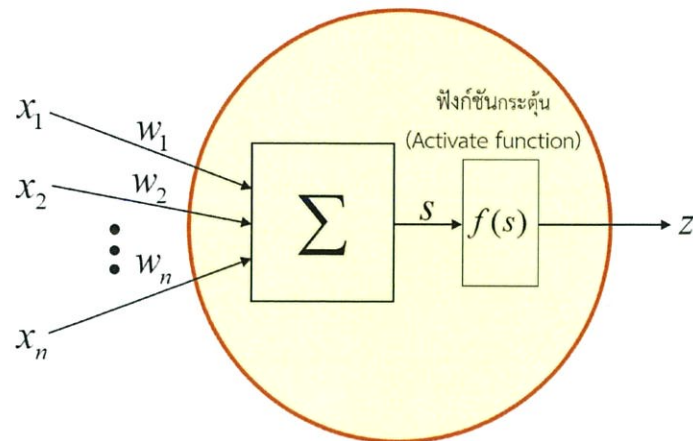
2.4.1 โครงข่ายประสาทเทียม (artificial neural network)

โครงข่ายประสาทเทียมเป็นหนึ่งในโมเดลการเรียนรู้ของเครื่อง (machine learning) เป็นส่วนหนึ่งของปัญญาประดิษฐ์ (artificial Intelligence) โครงข่ายประสาทเทียมมีความคิดริเริ่มมาจากการศึกษาโครงข่ายไฟฟ้าชีวภาพ (bioelectric network) ภายในสมอง ซึ่งประกอบด้วย 2 ส่วนคือ เซลล์ประสาท (neurons) และ จุดประสานประสาท (synapses) โครงข่ายประสาทเทียมสามารถช่วยในการแก้ปัญหา เช่น การจำแนกรูปภาพ และการประมวลผลภาษาธรรมชาติ (Natural Language Processing) เป็นต้น ซึ่งในกรณีจะใช้โครงข่ายประสาทเทียมแบบมีผู้สอน (artificial neural network supervised learning) โดยเป็นการให้โครงข่ายเรียนรู้จากชุดข้อมูลฝึกสอน (Training set) ที่ผู้สอนสร้างขึ้น ซึ่งประกอบด้วยข้อมูลอินพุตและค่าเอาต์พุตที่เป็นผลตอบสนองเป้าหมาย (Target response)

รูปแบบการฝึกสอนโครงข่ายประสาทเทียมแบบมีผู้สอน เริ่มต้นด้วยการป้อนอินพุตเข้าสู่โครงข่ายประสาทเทียม เพื่อให้โครงข่ายประสาทเทียมสร้างผลตอบสนองออกมาเป็นเอาต์พุต ซึ่งผลเอาต์พุตจะเป็นอย่างไร ขึ้นอยู่กับสถานะเริ่มต้นของโครงข่ายประสาทเทียม หลังจากนั้นเอาต์พุตดังกล่าวจะถูกนำไปเปรียบเทียบกับค่าเอาต์พุตเป้าหมายเพื่อหาความแตกต่าง ซึ่งความแตกต่างนั้นจะถูกพิจารณาเป็นความคลาดเคลื่อนที่นำไปคำนวณปรับค่าสถานะต่างๆ ในโครงข่ายประสาทเทียมให้ความคลาดเคลื่อนมีค่าน้อยที่สุด จากรูปแบบเบื้องต้นของการฝึกสอนโครงข่ายประสาทเทียมแบบมีผู้สอนสามารถอธิบายโดยรายละเอียดได้ 2 กระบวนการหลักคือ

2.4.1.1 กระบวนการส่งผ่าน

กระบวนการส่งผ่านเป็นกระบวนการได้มาซึ่งผลตอบสนองหรือเอาต์พุตของโครงข่ายประสาทเทียม โดยเริ่มพิจารณาจากโครงข่ายประสาทเทียม 1 หน่วย ดังรูปที่ 2.20



รูปที่ 2.20 แบบจำลองโครงข่ายประสาทเทียม 1 หน่วย

จากรูปที่ 2.20 อินพุตของประสาทเทียมหน่วยคือ x_1, x_2, \dots, x_n โดยอินพุตจะถูกคูณด้วยค่าถ่วงน้ำหนักของแต่ละอินพุตคือ w_1, w_2, \dots, w_n และเข้าสู่บล็อกผลรวม บล็อกผลรวมจะทำหน้าที่รวมผลคูณระหว่างถ่วงน้ำหนักกับอินพุตทั้งหมดได้ผลลัพธ์คือ s ซึ่งสามารถเขียนสมการได้ดังสมการที่ 2.11

$$s = x_1 w_1 + x_2 w_2 + \dots + x_n w_n \quad (2.11)$$

หรือ

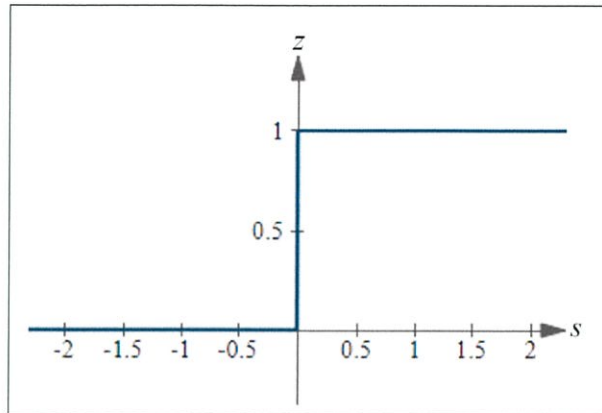
$$s = \sum_i^n x_i w_i \quad (2.12)$$

เมื่อได้ s แล้วกระบวนการต่อไปคือ กำหนดฟังก์ชันกระตุ้น (Activate function) เพื่อใช้จำกัดช่วงของเอาต์พุตที่จะเกิดขึ้นคือ z เนื่องจากเอาต์พุตที่ได้จะถูกส่งต่อไปยังโครงข่ายประสาทเทียมในชั้นอื่น ๆ ต่อไป โดยฟังก์ชันกระตุ้นแทนด้วย $f(s)$ แบ่งเป็น 4 ชนิด ได้ดังนี้ คือ

1. Threshold function สามารถแสดงดังสมการที่ 2.13

$$z = \begin{cases} 1, & s > 0 \\ 0, & s \leq 0 \end{cases} \quad (2.13)$$

จากสมการที่ 2.6 สามารถพล็อตกราฟได้ดังรูปที่ 2.21

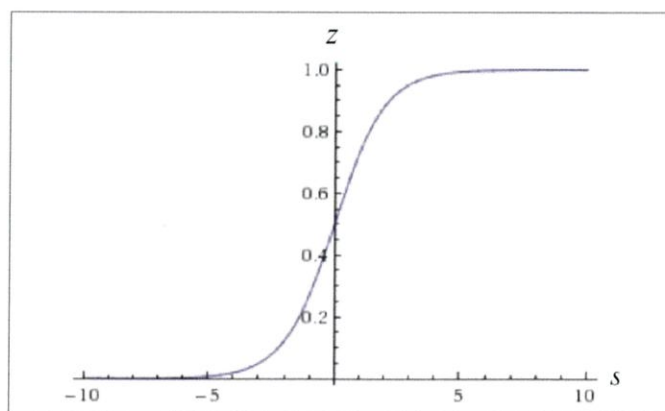


รูปที่ 2.21 กราฟผลตอบสนองของ Threshold function [8]

2. Logistic function เป็นฟังก์ชันที่ถูกใช้งานกับโครงข่ายประสาทเทียมอย่างแพร่หลาย เพราะ ผลตอบสนองของฟังก์ชันที่คล้ายกับ Threshold function แต่เป็นแบบต่อเนื่อง ซึ่งหมายความว่า เมื่ออินพุตของฟังก์ชันมีค่าน้อยผลตอบสนองฟังก์ชันจะเข้าใกล้ 0 แต่หากอินพุตของฟังก์ชันมีค่ามาก ผลตอบสนองฟังก์ชันจะมีค่าเข้าใกล้ 1 โดยสามารถเขียนสมการได้ในสมการที่ 2.14

$$z = \frac{1}{1 + e^{-s}} \quad (2.14)$$

จากสมการที่ 2.14 สามารถพล็อตกราฟได้ดังรูปที่ 2.22 โดยผลตอบสนองฟังก์ชันจะอยู่ในช่วง 0 ถึง 1 และมีรูปร่างคล้ายกับตัวอักษร S (S-shape)

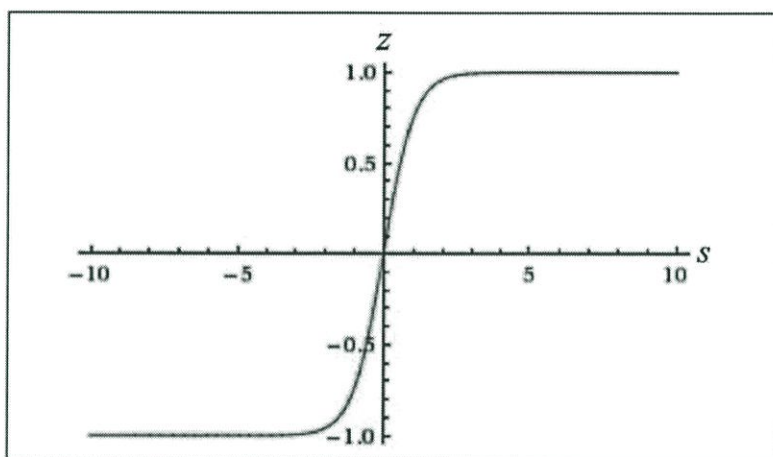


รูปที่ 2.22 กราฟผลตอบสนองของ sigmoid function [8]

3. hyperbolic tangent function เป็นฟังก์ชันที่คล้ายกับ sigmoid function แต่จุดกึ่งกลางของผลตอบสนองมีค่าเท่ากับ 0 ซึ่งสามารถเขียนสมการได้ในสมการที่ 2.15

$$z = \frac{e^s - e^{-s}}{e^s + e^{-s}} \quad (2.15)$$

จากสมการที่ 2.15 สามารถพล็อตกราฟได้ดังรูปที่ 2.23 พบว่ากราฟผลตอบสนองฟังก์ชันจะมีช่วงของผลตอบสนองอยู่ในช่วง -1 ถึง 1

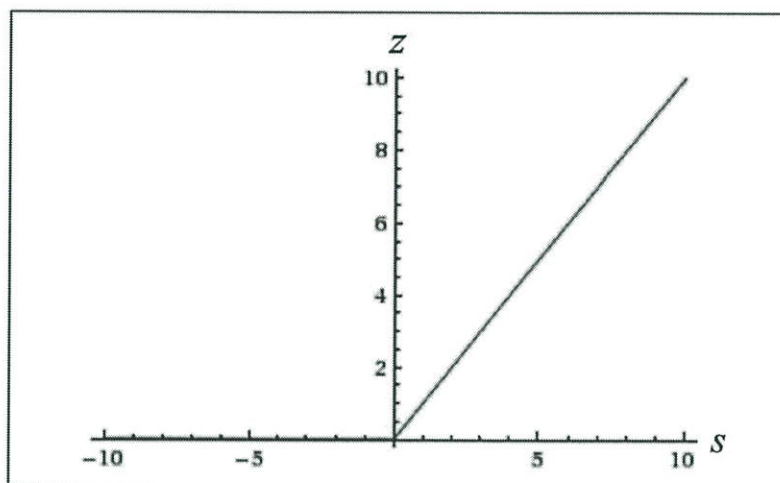


รูปที่ 2.23 กราฟผลตอบสนองของ hyperbolic tangent function [8]

4. restricted linear unit function เป็นฟังก์ชันที่ความแตกต่างกับฟังก์ชันที่เคยกล่าวมาเพราะฟังก์ชันจะคล้ายกับฟังก์ชันเชิงเส้น นอกจากนี้ยังเป็นฟังก์ชันที่เหมาะสมสำหรับการใช้งานเพื่อจำแนกภาพด้วยคอมพิวเตอร์วิทัศน์ (computer vision) ที่มีหลายปัจจัยอินพุตและหลายเอาต์พุต โดยสามารถเขียนสมการได้ในสมการที่ 2.16

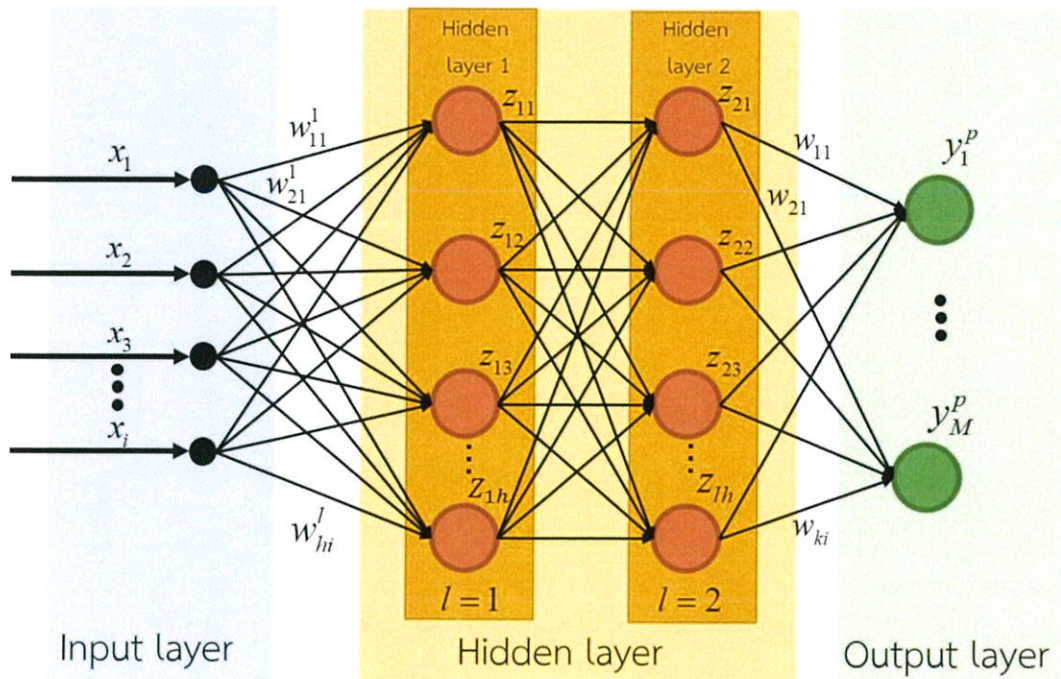
$$f(z) = \max(0, s) \quad (2.16)$$

จากสมการที่ 2.16 สามารถพล็อตกราฟได้ดังรูปที่ 2.24 ซึ่งมีรูปคล้ายกับไม้ฮอกกี้ โดยผลตอบสนองฟังก์ชันที่อินพุตต่ำกว่า 0 จะมีค่าเป็น 0 และเมื่ออินพุตมีค่ามากกว่า 0 จะให้ผลตอบสนองเท่ากับอินพุต



รูปที่ 2.24 กราฟผลตอบสนองของ restricted linear unit function [8]

การเลือกใช้งานฟังก์ชันกระตุ้นจะขึ้นอยู่กับความเหมาะสมของแต่ละชุดข้อมูลที่ใช้กับโครงข่ายประสาทเทียม กระบวนการทั้งหมดที่กล่าวมาข้างต้นจะเป็นกระบวนการภายในโครงข่ายประสาทเทียม 1 หน่วย หรือเรียกได้ว่า 1 โหนด (Node) โครงข่ายประสาทเทียมที่ใช้งานในปัจจุบันเป็นสถาปัตยกรรมโครงข่ายประสาทเทียมเพอร์เซพตรอนแบบหลายชั้น (multilayer perceptron) กล่าวคือ สถาปัตยกรรมโครงข่ายจะแบ่งเป็น 3 ส่วนคือ ชั้นอินพุต ชั้นซ่อน และชั้นเอาต์พุต ชั้นอินพุตเป็นชั้นที่รับค่าอินพุตเข้าสู่โครงข่ายไม่มีการคำนวณทางคณิตศาสตร์แต่เป็นจุดส่งต่อไปยังชั้นต่อไปยังชั้นซ่อน โดยชั้นซ่อนถูกกำหนดให้มีมากกว่า 1 ชั้น และในแต่ละชั้นสามารถกำหนดจำนวนโหนดซ่อนได้ ในชั้นสุดท้ายเป็นชั้นเอาต์พุตมีเพียงชั้นเดียว ซึ่งโหนดเอาต์พุตจะถูกกำหนดตามเอาต์พุตที่ต้องการตามวัตถุประสงค์การใช้งาน สามารถแสดงตัวอย่างสถาปัตยกรรมโครงข่ายประสาทเทียมเพอร์เซพตรอนแบบหลายชั้นได้ดังรูปที่ 2.25



รูปที่ 2.25 สถาปัตยกรรมโครงข่ายประสาทเทียมเพอร์เซพตรอนแบบหลายชั้น

จากรูปที่ 2.25 พบว่าในชั้นซ่อนและชั้นเอาต์พุตมีจำนวนโนดมากกว่า 1 โนด ทำให้ต้องมีการปรับปรุงการทำงานทางคณิตศาสตร์จากสมการที่ 2.5 โดยมีการเพิ่มค่าไบอัส (bias) เพื่อระบุความสำคัญของแต่ละโนดเพิ่มเติม และมีการรับส่งข้อมูลอินพุตกับเอาต์พุตเชื่อมต่อกันในแต่ละโนด จึงสามารถเขียนเป็นสมการได้ดังสมการที่ 2.17

$$s_{lh} = \sum_{i=1}^n w_{hi}^l x_i + b_{lh} \quad (2.17)$$

- เมื่อ l แทนลำดับชั้นของชั้นซ่อน
 h แทนลำดับโนดในแต่ละชั้นซ่อน
 i แทนลำดับของอินพุต
 n แทนจำนวนของอินพุต
 s_{lh} แทนผลลัพธ์การคูณอินพุตกับค่าถ่วงน้ำหนักและบวกค่าไบอัสในโนดซ่อน
 w_{hi}^l แทนค่าถ่วงน้ำหนักในแต่ละโนดในชั้นซ่อน
 b_{lh} แทนค่าไบอัสในแต่ละโนดในชั้นซ่อน
 x_i แทนอินพุตของแต่ละโนดในชั้นซ่อน

จากสมการที่ 2.17 เมื่อได้ค่า s_{jh} ในแต่ละโนดต้องผ่านฟังก์ชันกระตุ้นชนิดเดียวกันทั้งโครงข่ายซึ่งสามารถกำหนดได้ จากทั้งหมด 4 ชนิดที่กล่าวไปข้างต้นแล้วซึ่งแสดงเป็นสมการได้ดังสมการที่ 2.18

$$z_{jh} = f(s_{jh}) \quad (2.18)$$

เมื่อ z_{jh} แทนผลลัพธ์ในแต่ละโนดในชั้นซ่อน
 s_{jh} แทนผลผลลัพธ์ของการคูณอินพุตกับค่าถ่วงน้ำหนักและไปอัส
 $f(s_{jh})$ แทนฟังก์ชันกระตุ้น

จากสมการที่ 2.17 และ 2.18 เป็นกระบวนการทางคณิตศาสตร์แต่ละโนดในชั้นซ่อนซึ่งเป็นกระบวนการเดียวกันกับโนดในชั้นเอาต์พุต โดยเปลี่ยนตัวแปรในการคำนวณดังสมการที่ 2.19

$$s_k^p = \sum_{i=1}^n w_{ki} x_i + b_k \quad (2.19)$$

เมื่อ k แทนลำดับโนดเอาต์พุต
 i แทนลำดับของอินพุตจากชั้นซ่อน
 n แทนจำนวนของอินพุตจากชั้นซ่อน
 s_k^p แทนผลลัพธ์ของการคูณอินพุตกับค่าถ่วงน้ำหนักและบวกด้วยไปอัสในโนดเอาต์พุต
 w_{ki} แทนผลผลลัพธ์ของแต่ละโนด
 b_k แทนผลค่าไปอัสในแต่ละโนดในชั้นเอาต์พุต

หลังจากนั้นนำค่า s_k^p เข้าสู่ฟังก์ชันกระตุ้นดังสมการที่ 2.8 โดยกำหนดให้ s_{jh} แทนด้วย s_k^p และ z_{jh} แทนด้วย y_k^p ซึ่งเป็นผลลัพธ์ของแต่ละโนดเอาต์พุต

เมื่อได้ค่า y_k^p ถือเป็นผลลัพธ์ของสถาปัตยกรรมโครงข่ายประสาทเทียมเพอร์เซพตรอนแบบหลายชั้น และเป็นการสิ้นสุดกระบวนการส่งผ่าน ซึ่งค่า y_k^p ในสถานะเริ่มต้นของโครงข่ายจะยังคงไม่เท่ากับค่าเป้าหมายที่ต้องการ ดังนั้นจึงต้องอาศัยการกระบวนการย้อนกลับเพื่อปรับสถานะในโครงข่ายให้ได้ตามค่าเป้าหมาย โดยการปรับค่าถ่วงน้ำหนักภายในโครงข่าย

2.4.1.2 กระบวนการย้อนกลับ

ในกระบวนการย้อนกลับมีเป้าหมายเพื่อลดค่าความผิดพลาดที่เกิดความแตกต่างระหว่างเอาต์พุตเป้าหมายกับเอาต์พุตที่ได้จากกระบวนการส่งผ่าน โดยคำนวณหาความผิดพลาดจากสมการที่ 2.20

$$E = \frac{1}{2} \sum_{k=1}^M (y_k^t - y_k^p)^2 \quad (2.20)$$

เมื่อ	M	แทนจำนวนโนดเอาต์พุต
	k	แทนลำดับโนดเอาต์พุต
	y_k^p	แทนผลลัพธ์ของแต่ละโนดเอาต์พุต
	y_k^t	แทนเอาต์พุตเป้าหมายแต่ละโนดเอาต์พุต
	E	แทนค่าความผิดพลาด

เมื่อได้ E นำค่าเข้าสู่กระบวนการเคลื่อนลงตามความชัน (gradient descent) ซึ่งเป็นหนึ่งในกระบวนการหาค่าเหมาะสมที่สุด ซึ่งในกรณีนี้ต้องการหาค่าต่ำสุดของ E โดยกระบวนการเคลื่อนลงตามความชันในการปรับค่าถ่วงน้ำหนักสามารถเขียนเป็นสมการได้ดังสมการที่ 2.21

$$w^+ = w - \eta \frac{\partial E}{\partial w} \quad (2.21)$$

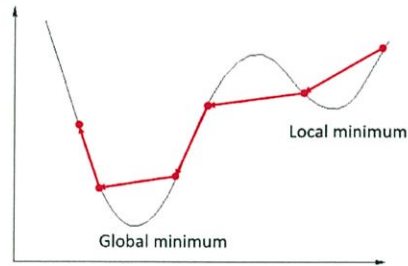
เมื่อ	w^+	แทนค่าถ่วงน้ำหนักที่ถูกปรับ
	w	แทนค่าถ่วงน้ำหนักเริ่มต้น
	E	แทนค่าความผิดพลาดเฉลี่ย
	η	แทนอัตราการเรียนรู้

จากสมการที่ 2.21 ค่าถ่วงน้ำหนักภายในโครงข่ายประสาทเทียมจะถูกปรับ โดยอัตราการปรับจะขึ้นอยู่กับอัตราการเรียนรู้ เมื่อเสร็จสิ้นกระบวนการปรับค่าถ่วงน้ำหนักก็จะเป็นการสิ้นสุดกระบวนการย้อนกลับ โดยนับเป็น 1 รอบการเรียนรู้ (iterations)

ในกระบวนการย้อนกลับตัวแปรสำคัญคือ อัตราการเรียนรู้ซึ่งเป็นตัวแปรที่กำหนดจากตำแหน่งความชันเดิมไปยังความชันใหม่และด้วยเหตุนี้ทำให้การกำหนดค่าอัตราการเรียนรู้เป็นที่เหมาะสมเป็นเรื่องยาก เนื่องจากหากกำหนดค่าน้อยเกินไปจะทำให้ต้องใช้เวลาในการคำนวณซ้ำหลายครั้ง และมีโอกาสสูงมากที่จุดของความชันที่พบเป็นเพียงจุดต่ำสุดสัมพัทธ์ แต่ถ้ากำหนดค่ามากเกินไปก็จะทำให้ถูกปรับข้ามจุดต่ำสุดสัมบูรณ์ หรือจุดที่เหมาะสม ซึ่งสามารถแสดงได้ดังรูปที่ 2.26

Gradient Descent

Big learning rate



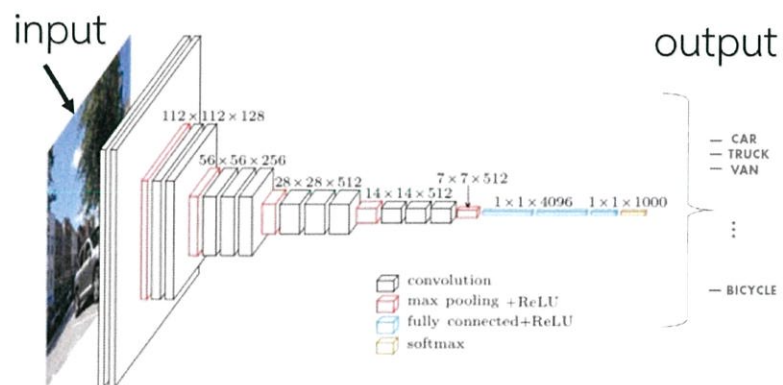
Small learning rate



รูปที่ 2.26 เปรียบเทียบผลกระทบบจากการปรับอัตราการเรียนรู้ (learning rate) [9]

2.4.2 Convolutional Neural Network

Convolutional Neural Network เป็นการนำคอนโวลูชันแบบ 2 มิติโดยให้ตัวกรองเป็นตัวแปรค่าถ่วงน้ำหนักและลดจำนวนพิกเซลของลงด้วยกระบวนการ pooling ซึ่งโครงสร้างของ Convolutional Neural Network โดยการทำงานเป็นดังรูปที่ 2.27



รูปที่ 2.27 ภาพรวมการทำงานของ Convolutional Neural Network [9]

การทำงานของ Convolutional Neural Network สามารถแบ่งได้เป็น 4 ส่วน
ดังต่อไปนี้

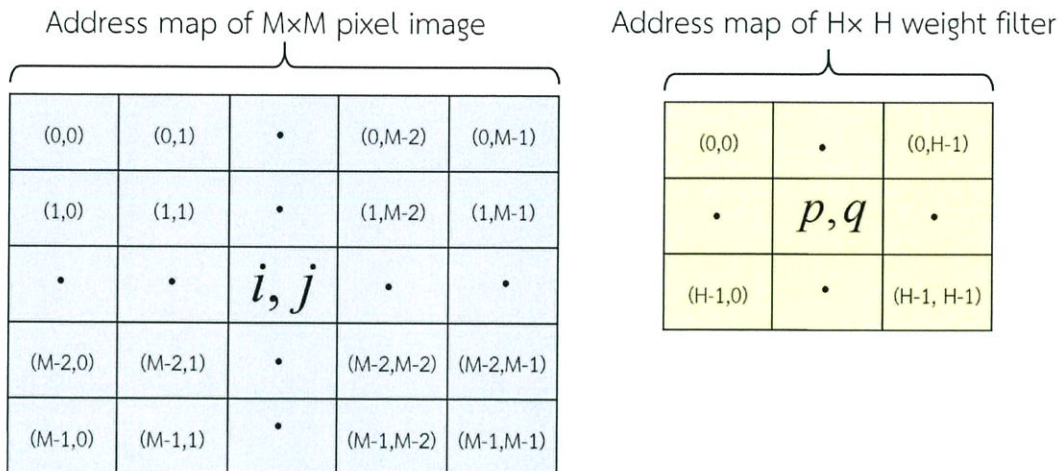
2.4.2.1 Convolution layer

Convolution layer เป็นการนำรูปมาคอนโวลูชัน 2 มิติกับตัวกรองหลายตัวในลักษณะต่อแบบขนานซึ่งสามารถทำได้ทั้งภาพสีและภาพขาวดำ ซึ่งสมการของการหาเอาต์พุตในละตัวกรองสามารถเขียนได้ดังสมการที่ 2.22

$$u_{i,j,m} = \sum_{k=0}^{K-1} \sum_{p=0}^{H-1} \sum_{q=0}^{H-1} z_{i+p,j+q,k}^{(l-1)} h_{p,q,k,m} + b_m \quad (2.22)$$

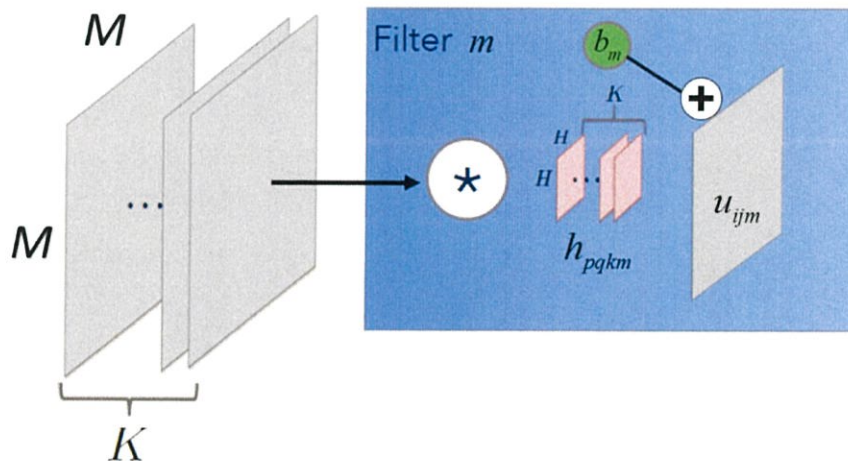
เมื่อ	K	แทนชั้นของภาพอินพุต เช่นภาพสีเท่ากับ 3 (RGB) ภาพขาวดำ (grayscale) เท่ากับ 1
	H	แทนขนาดของตัวกรอง
	i	แทนลำดับพิกเซลของภาพในแนวแกนนอน
	j	แทนลำดับพิกเซลของภาพในแนวแกนตั้ง
	p	แทนลำดับค่าถ่วงน้ำหนักในแนวแกนนอน
	q	แทนลำดับค่าถ่วงน้ำหนักในแนวแกนตั้ง
	m	แทนลำดับของตัวกรอง
	l	แทนลำดับของการคอนโวลูชัน
	z^{l-1}	ทนค่าแต่ละพิกเซลของภาพอินพุต
	h_{pqkm}	แทนค่าถ่วงน้ำหนักในแต่ละตำแหน่งในตัวกรอง
	b_m	แทนค่าไบอัสในแต่ละตัวกรอง
	$u_{i,j,m}$	แทนผลลัพธ์ของชั้น Convolution layer

จากสมการที่ 2.22 สามารถอธิบายตัวแปรตำแหน่งต่าง ๆ ในภาพอินพุตและตัวกรองถ่วงน้ำหนักได้ดังรูปที่ 2.28



รูปที่ 2.28 ตัวแปรตำแหน่งภาพอินพุตและตัวกรองถ่วงน้ำหนัก

จากสมการที่ 2.15 สามารถอธิบายตัวแปรในกระบวนการคอนโวลูชันได้ดังรูปที่ 2.29



รูปที่ 2.29 กระบวนการใน convolution layer

ดังนั้นเมื่อทำการคอนโวลูชันแต่ละตัวกรองแบบขนานจะทำให้ได้ผลลัพธ์เป็นภาพขนานเท่าภาพต้นฉบับตามจำนวนของตัวกรอง

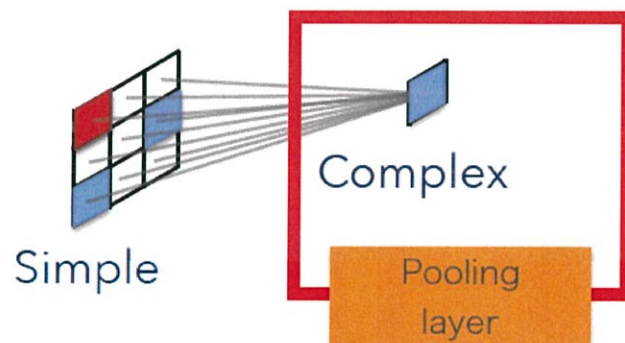
2.4.2.2 Pooling layer

Pooling layer เป็นการลดอัตราสุ่มลง คือการลดขนาดของภาพลงซึ่งต้องกำหนดขนาดกรอบที่จะลดพิกเซลลงเรียกว่า Pooling size และขั้นในการเลื่อนกรอบเรียกว่า Stride เมื่อกำหนดค่าทั้งสองจะสามารถคำนวณหาขนาดของภาพเอาต์พุตได้ด้วยสมการที่ 2.23

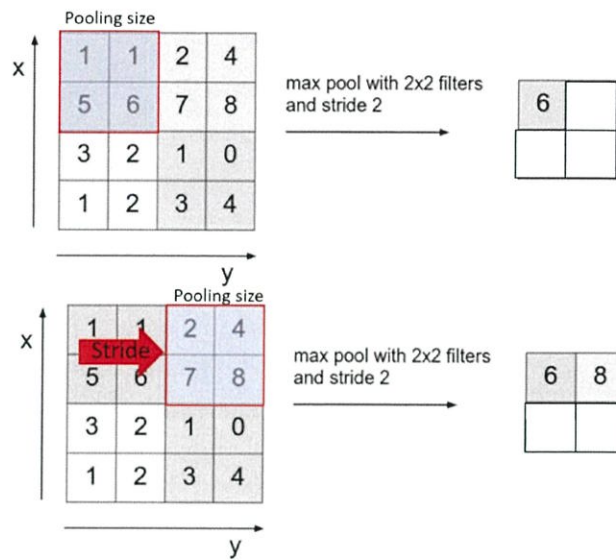
$$N = \lfloor (M-1)/s \rfloor + 1 \quad (2.23)$$

เมื่อ M แทนขนาดภาพอินพุต
 N แทนขนาดภาพเอาต์พุต
 s ชั้นในการเลื่อนกรอบ Stride

กระบวนการทำงานในชั้น pooling layer สามารถอธิบายได้โดยรูปที่ 2.30 Pooling layer จะครอบกรอบ Pooling size จะถูกลดขนาดภาพที่ถูกลดอัตราส่วนลงจากทั้งหมดในกรอบเหลือเพียง 1 ค่า หรือ 1 พิกเซล ซึ่งหลักการสามารถกำหนดได้ดังนี้ การใช้วิธีค่าเฉลี่ยของพิกเซลทั้งหมดภายในกรอบ (Average Pool) การใช้ค่ามากที่สุดในกรอบ (Max Pool) หลังจากนั้นจะเลื่อนกรอบตามขั้นเท่ากับค่า Stride ที่กำหนดไว้ โดยเลื่อนตำแหน่งซ้ายไปขวาและบนลงล่าง ดังรูปที่ 2.31 ซึ่งเป็นการแสดงผลการทำ Max Pool โดย Pooling size เท่ากับ 2×2 และ Stride เท่ากับ 2



รูปที่ 2.30 กระบวนการทำงานในชั้น pooling layer [10]

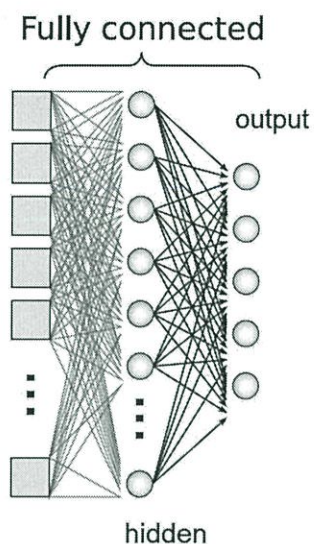


รูปที่ 2.31 การลดอัตราสุ่มแบบ Average Pooling [10]

เมื่อผ่านกระบวนการผลลัพธ์ที่ได้จะเข้าสู่ฟังก์ชันกระตุ้นซึ่งในกระบวนการ Convolutional Neural Network นิยมใช้ฟังก์ชันกระตุ้นแบบ restricted linear unit function

2.4.2.3 Fully-connected layer

Fully-connected layer เป็นการนำภาพที่ได้กระจายออกเป็นจาก 2 มิติเป็น 1 มิติ และเพื่อเข้าเป็นอินพุตของโครงข่ายประสาทเทียม ซึ่งตัวแปรที่กำหนดคือจำนวนของเซลล์ประสาทในชั้นซ่อนตัวของโครงข่ายประสาทเทียมและกำหนดเอาต์พุตของชั้นการทำงาน ซึ่งสามารถแสดงได้ดังรูปที่ 2.32



รูปที่ 2.32 การทำงานใน Fully-connected layer [11]

ซึ่งกระบวนการนี้จะเป็นการเปลี่ยนลักษณะเฉพาะในแต่ละตัวกรองถ่วงน้ำหนักหรือเรียกได้ว่าเป็นตัวกรองภาพเข้าสู่กระบวนการจำแนกด้วยโครงข่ายประสาทเทียมอีกครั้งหนึ่ง โดยมีกเป็นขั้นสุดท้ายของการทำ Convolutional Neural Network ก่อนเข้าฟังก์ชัน softmax

2.4.2.4 softmax

Softmax เป็นฟังก์ชันหาความน่าจะเป็นของอินพุต ซึ่งอินพุตเป็นชุดข้อมูลเอาต์พุตจากชั้น fully-connected layer เนื่องจากค่าที่ได้จาก fully-connected layer เป็นค่าตัวเลขทศนิยมหลายตำแหน่ง ซึ่งมีค่าน้อยมากแตกต่างกันไปตามอินพุตของ Convolutional Neural Network ดังนั้นเพื่อให้ง่ายต่อการสร้างเงื่อนไขในตัดสินใจจำแนกอินพุตที่ได้รับมา จึงต้องหาความน่าจะเป็นของชุดข้อมูลก่อนตัดสินใจ โดยเลือกจากค่าความน่าจะเป็นสูงที่สุด โดยสามารถเขียนเป็นสมการได้ดังสมการที่ 2.17

$$Sf(y_i) = \frac{e^{y_i}}{\sum_{i=1}^I e^{y_i}} \quad (2.24)$$

เมื่อ	y_i	แทนค่าอินพุต
	$Sf(y_i)$	แทนความน่าจะเป็นของค่าอินพุต
	i	แทนลำดับของอินพุต
	I	แทนจำนวนของอินพุต

เมื่อได้เอาต์พุตของฟังก์ชัน Softmax เป็นการจบกระบวนการทำงานในของ Convolutional Neural Network

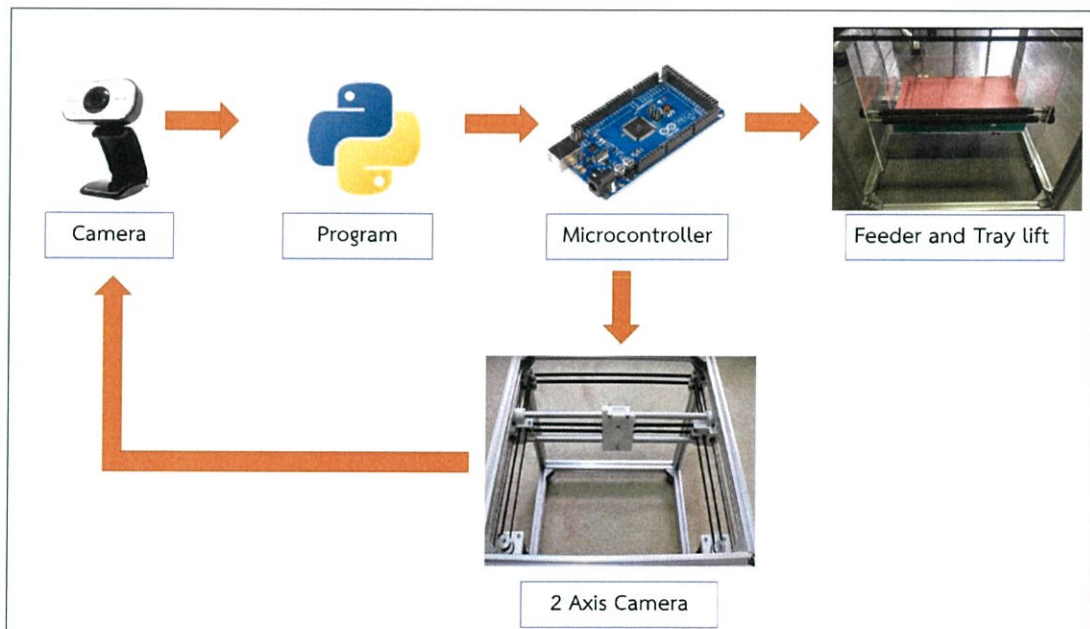
ซึ่งกระบวนการเรียนรู้ย้อนกลับของ Convolutional Neural Network มีหลักการเดียวกับโครงข่ายประสาทเทียมดังที่อธิบายไปแล้ว ซึ่งการแทนค่า y_k^p ในสมการที่ 2.13 จะใช้ค่าที่ได้จากเอาต์พุตจากชั้น fully-connected layer

บทที่ 3

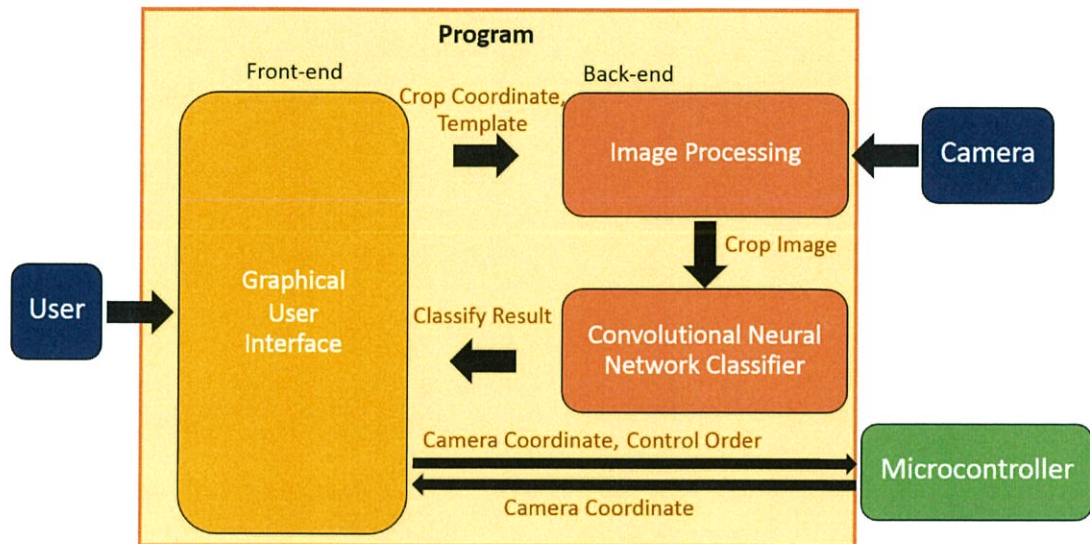
การออกแบบและการจัดทำปฏิญญาฉบับ

3.1 การออกแบบ

ระบบรวมคะแนนข้อสอบแบบอัตโนมัติโดยใช้ภาษาไพธอน (Python) ในการพัฒนาระบบ และใช้ไมโครคอนโทรลเลอร์ (Microcontroller) ในการควบคุมกลไกการเปลี่ยนชุดข้อสอบและการเลื่อนตัวกล้องสองแกน ซึ่งกระบวนการทำงานของระบบเริ่มจากโปรแกรมรับภาพจากกล้องเพื่อประมวลผลในคอมพิวเตอร์ โดยทำการตัดกรอบและจำแนกตัวเลขโดยใช้ข้อมูลจากกระบวนการเรียนรู้และจดจำ Convolutional neural network (CNN) หลังจากการประมวลผลเสร็จสิ้นจะส่งต่อการทำงานไปยังไมโครคอนโทรลเลอร์ที่ควบคุมตัวป้อนกระดาษ (Feeder and Tray lift) เพื่อเปลี่ยนชุดข้อสอบและยกถาดรองชุดข้อสอบขึ้นให้พอดีกับลูกยางป้อนกระดาษ เมื่อผู้ใช้ต้องการเก็บข้อมูลอื่น ๆ สามารถส่งคำสั่งไปยังไมโครคอนโทรลเลอร์ให้ควบคุมชุดเลื่อนตัวกล้องสองแกน (2 Axis Camera) ไปยังตำแหน่งที่ต้องการตรวจจับภาพบนชุดข้อสอบ แสดงบล็อกไดอะแกรมของโครงการได้ดังรูปที่ 3.1 แสดงบล็อกไดอะแกรมของโครงการได้ดังรูปที่ 3.2



รูปที่ 3.1 บล็อกไดอะแกรมของระบบรวมคะแนนข้อสอบแบบอัตโนมัติ



รูปที่ 3.2 บล็อกไดอะแกรมของโปรแกรม

การทำงานของระบบรวมคะแนนข้อสอบอัตโนมัติสามารถแบ่งส่วนการออกแบบเป็น 5 ส่วนคือ

1. ชุดเคลื่อนตัวกล้องสองแกนและโปรแกรมควบคุม
2. เครื่องเปลี่ยนชุดข้อสอบและโปรแกรมควบคุม
3. โปรแกรมประมวลผลภาพ
4. โปรแกรมจำแนกตัวเลขด้วยกระบวนการเรียนรู้และจดจำแบบ CNN
5. โปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ (Graphical User Interface : GUI)

ซึ่งการออกแบบการทำงานในแต่ละส่วนประกอบด้วยซอฟต์แวร์ และฮาร์ดแวร์ซึ่งสามารถอธิบายโดยละเอียดดังนี้

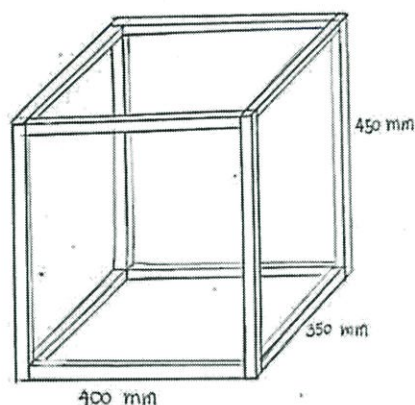
3.1.1 การออกแบบชุดเคลื่อนตัวกล้องสองแกนและซอฟต์แวร์ควบคุม

3.1.1.1 โครงสร้าง

การออกแบบโครงสร้างชุดเคลื่อนตัวกล้องสองแกน มีโครงสร้างหลักเป็นอลูมิเนียมโปรไฟล์ โดยรายละเอียด และวิธีการประกอบแสดงเป็นลำดับดังต่อไปนี้

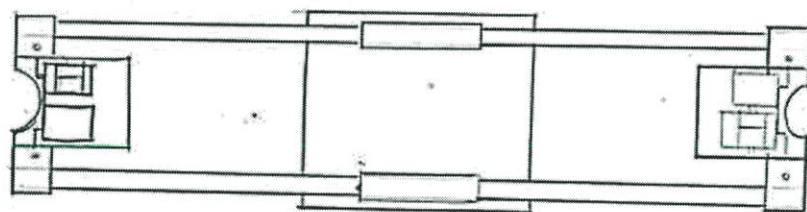
1. การขึ้นโครงอลูมิเนียมโปรไฟล์ (Aluminum Profile) โดยนำอลูมิเนียมโปรไฟล์ที่มีความยาว 400 mm เป็นด้านยาว ชั้นที่มีความยาว 350 mm เป็นด้านกว้าง และชั้นที่มีความยาว 450

mm เป็นชิ้นด้านสูง แล้วใช้ฉากอลูมิเนียมเป็นตัวยึดอลูมิเนียมโปรไฟล์ให้ติดกันที่บริเวณมุมข้อต่อของโครงสร้าง ลักษณะการขึ้นโครงอลูมิเนียมโปรไฟล์เทียบกับที่ออกแบบไว้ดังภาพที่ 3.3



รูปที่ 3.3 ลักษณะการขึ้นโครงอลูมิเนียมโปรไฟล์ที่ออกแบบไว้

2. การประกอบอุปกรณ์ชุดเลื่อนล้อสองแกนเคลื่อนที่ในแนวแกน X โดยนำท่อตรง (linear tube) ที่นำแบริ่งลูกกลิ้งตรง (linear ball bearing) สวมเข้าไปทั้งสองแห่งของท่อเชิงเส้น แล้วนำมาติดตั้งกับชิ้นส่วน X Carriage โดยให้ชิ้นส่วน X Carriage สวมครอบกับแบริ่งลูกกลิ้งตรงทั้งสอง และนำชิ้นส่วน XY Joiner มาติดตั้งกับท่อตรงที่บริเวณปลายทั้งสองข้างของท่อเชิงเส้น ลักษณะการสร้างอุปกรณ์ชุดเลื่อนล้อสองแกนเคลื่อนที่ในแนวแกน X เทียบกับที่ออกแบบไว้ดังภาพที่ 3.4



รูปที่ 3.4 การประกอบอุปกรณ์ชุดเลื่อนตัวล้อสองแกนเคลื่อนที่ในแนวแกน X ที่ออกแบบไว้

3. การประกอบอุปกรณ์ชุดเลื่อนล้อสองแกนเคลื่อนที่ในแนวแกน Y โดยนำเพลาตรง (linear shaft) ที่นำแบริ่งลูกกลิ้งตรงสวมเข้าไปทั้งสองแห่ง แล้วมาติดตั้งเข้ากับโครงอลูมิเนียมโปรไฟล์ที่ตำแหน่งส่วนบนสุดของโครงอลูมิเนียมโปรไฟล์ โดยใช้ชิ้นส่วน Y Shaft Clamp เป็นตัวยึดเพลาตรงกับโครงอลูมิเนียมโปรไฟล์เข้าด้วยกัน ลักษณะการติดตั้งเพลาตรง (linear shaft) ในแนวแกน Y แสดงดังภาพที่ 3.5



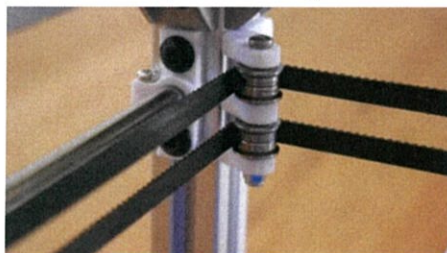
รูปที่ 3.5 การประกอบอุปกรณ์ชุดเลื่อนกลิ้งสองแกนเคลื่อนที่ในแนวแกน Y

4. การรวมอุปกรณ์ชุดเลื่อนกลิ้งสองแกนเคลื่อนที่ในแนวแกน X และ Y โดยนำชิ้นส่วน XY Joiner ที่ประกอบกับอุปกรณ์การเคลื่อนที่ในแนวแกน X มาสวมครอบแปรงลูกกลิ้ง ในแนวแกน Y ร่วมกับชิ้นส่วน XY Clamp ทำแบบเดียวกันทั้งสองด้าน ลักษณะการรวมอุปกรณ์ชุดเลื่อนกลิ้งสองแกนเคลื่อนที่ในแนวแกน X และ Y ดังรูปที่ 3.6



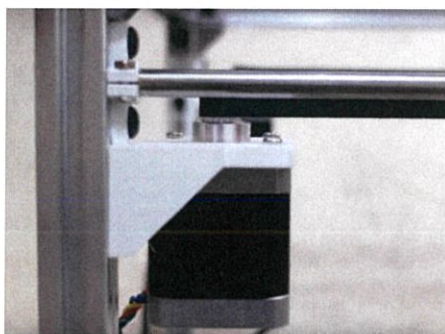
รูปที่ 3.6 ลักษณะการรวมอุปกรณ์ชุดเลื่อนกลิ้งสองแกนเคลื่อนที่ในแนวแกน X และ Y

5. ในส่วนการติดตั้งชิ้นส่วน Idler และมอเตอร์ ของชุดชุดเลื่อนกลิ้งสองแกนโดยนำลูกปืนตลับขนาดเล็ก (small ball bearing) มาติดตั้งเข้ากับชิ้นส่วน Idler จากนั้นนำชิ้นส่วน Idler มาติดตั้งเข้ากับโครงอลูมิเนียมโปรไฟล์ที่ตำแหน่งบนสุดของโครงอลูมิเนียมโปรไฟล์ ดังรูปที่ 3.7



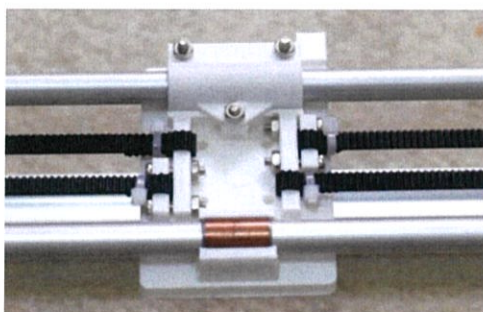
รูปที่ 3.7 การติดตั้งชิ้นส่วน Idler

จากนั้นนำชิ้นส่วนมอเตอร์มาติดตั้งเข้ากับชิ้นส่วน XY มอเตอร์จากนั้นนำไปติดตั้งกับโครงอลูมิเนียมโปรไฟล์ ดังรูปที่ 3.8



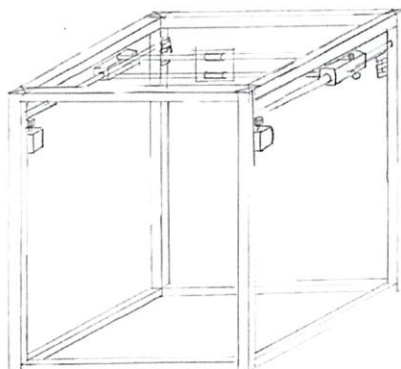
รูปที่ 3.8 การติดตั้งมอเตอร์

6. การติดตั้งสายพาน จะใช้สายพาน 2 เส้นขนาดเท่ากัน โดยใช้ปลายสายพานเส้นแรกยึดกับชิ้นส่วน X Carriage ฝั่งบนซ้าย ส่วนปลายอีกด้านยึดกับชิ้นส่วน X Carriage ฝั่งล่างขวา และใช้ปลายสายพานเส้นที่สองยึดกับชิ้นส่วน X Carriage ฝั่งล่างซ้าย ส่วนปลายอีกด้านยึดกับชิ้นส่วน X Carriage ฝั่งบนขวา ดังรูปที่ 3.9



รูปที่ 3.9 การยึดสายพานกับชิ้นส่วน X Carriage

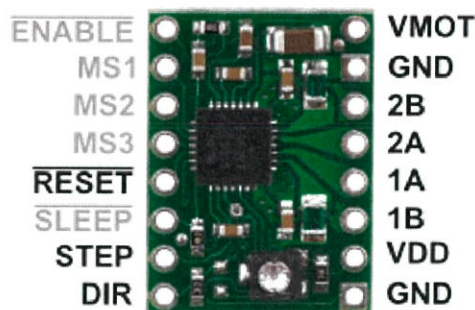
จากนั้นทำการวางตำแหน่งสายพานผ่านอุปกรณ์ และชิ้นส่วนต่างๆ เทียบกับที่ออกแบบไว้ตามลักษณะที่แสดงดังรูปที่ 3.10



รูปที่ 3.10 การวางตำแหน่งสายพานผ่านอุปกรณ์ และชิ้นส่วนต่าง ๆ ที่ออกแบบไว้

3.1.1.2 การเชื่อมต่อวงจร

การออกแบบวงจรควบคุมการทำงานของชุดเคลื่อนตัวกลิ้งสองแกนมีอุปกรณ์คือ Arduino Mega 2560 A4988 Stepper Driver Endstop switch และ Stepping Motor โดยการควบคุมชุดเคลื่อนตัวกลิ้งสองแกนจะใช้ A4988 Stepper Driver และ Stepping Motor ทั้งหมด 2 ชุด ซึ่งจะแบ่งเป็นชุด A และชุด B และใช้งาน Endstop switch 2 ตัวในระบบตั้งค่าเริ่มต้นของชุดเคลื่อนตัวกลิ้งสองแกน ซึ่งจะกำหนด Endstop switch 2 ตัวเป็น Endstop switch A และ Endstop switch B โดยรายละเอียดพินเชื่อมต่อของ A4988 Stepper Driver แสดงดังรูปที่ 3.10 และรายละเอียดพินเชื่อมต่อของ Endstop switch แสดงดังรูปที่ 3.11

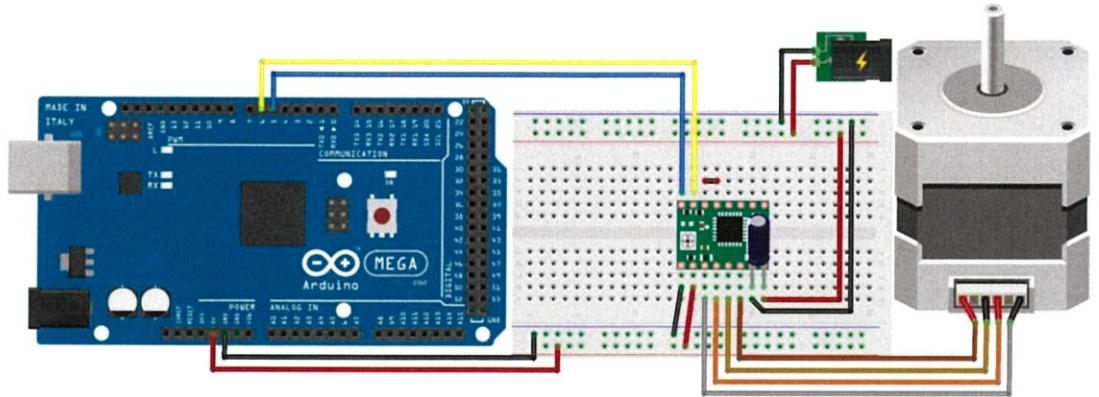


รูปที่ 3.11 รายละเอียดช่องเชื่อมต่อของ A4988 Stepper Driver [12]

จากรูปที่ 3.11 เป็นรายละเอียดชนิดของช่องเชื่อมต่อของ A4988 Stepper Driver ซึ่งการเชื่อมต่อขาแต่ละขาของ A4988 Stepper Driver ดังนี้

1. เชื่อมต่อพิน STEP และ DIR ของ A4988 Stepper Driver เข้ากับพิน 6 และ 5 ของ Arduino ตามลำดับ
2. เชื่อมต่อพิน 1B 1A 2A และ 2B ของ A4988 Stepper Driver เข้ากับสายที่ 1 2 3 และ 4 ของ Stepping Motor ตามลำดับ
3. เชื่อมต่อพิน RESET และ SLEEP ของ A4988 Stepper Driver เข้าด้วยกัน
4. เชื่อมต่อพิน 5V ของ Arduino เข้ากับพิน VCC ของ A4988 Stepper Driver
5. นำตัวเก็บประจุขนาด 100 μ F เชื่อมต่อขาบวก และขาลบเข้ากับพิน VMOT และ GND ของ A4988 Stepper Driver ตามลำดับ
6. เชื่อมต่อไฟเลี้ยง +12V เข้ากับพิน VMOT ของ A4988 Stepper Driver
7. เชื่อมต่อกราวด์ GND ของ Arduino และ GND ของไฟเลี้ยงเข้ากับพอร์ต power GND

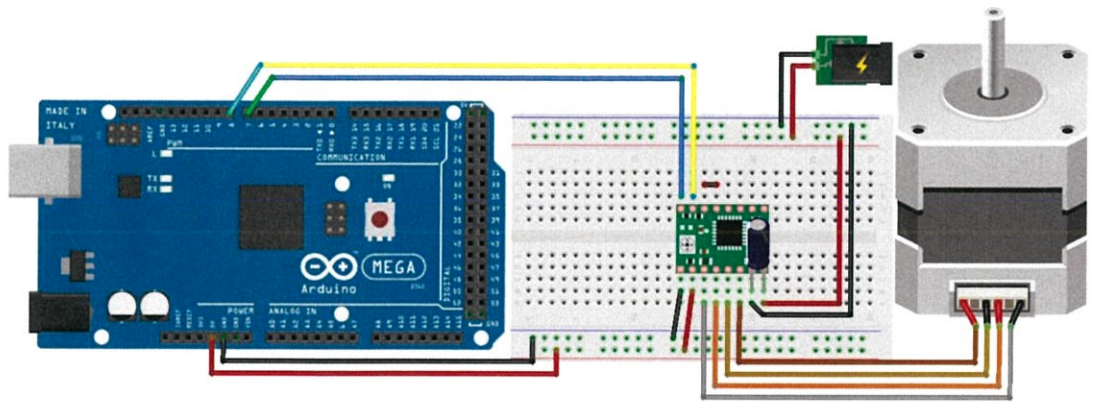
การเชื่อมต่อ Arduino A4988 Stepper Driver และ Stepping Motor ชุด A แสดง
 ดังรูปที่ 3.12



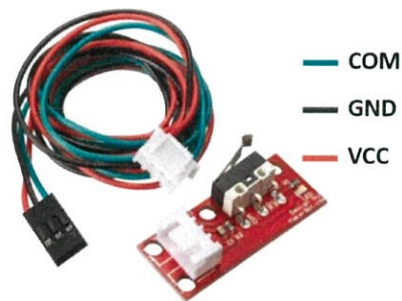
รูปที่ 3.12 การเชื่อมต่อระหว่าง Arduino A4988 Stepper Driver และ Stepping Motor ชุด A

1. เชื่อมต่อพิน STEP และ DIR ของ A4988 Stepper Driver เข้ากับพิน 6 และ 5 ของ Arduino ตามลำดับ
2. เชื่อมต่อพิน 1B 1A 2A และ 2B ของ A4988 Stepper Driver เข้ากับสายที่ 1 2 3 และ 4 ของ Stepping Motor ตามลำดับ
3. เชื่อมต่อพิน RESET และ SLEEP ของ A4988 Stepper Driver เข้าด้วยกัน
4. เชื่อมต่อพิน 5V ของ Arduino เข้ากับพิน VCC ของ A4988 Stepper Driver
5. นำตัวเก็บประจุขนาด 100 μ F เชื่อมต่อขาบวก และขาลบเข้ากับพิน VMOT และ GND ของ A4988 Stepper Driver ตามลำดับ
6. เชื่อมต่อไฟเลี้ยง +12V เข้ากับพิน VMOT ของ A4988 Stepper Driver
7. เชื่อมต่อกราวด์ GND ของ Arduino และ GND ของไฟเลี้ยงเข้ากับพอร์ต power GND

การเชื่อมต่อ Arduino A4988 Stepper Driver และ Stepping Motor ชุด B แสดง
 ดังรูปที่ 3.13



รูปที่ 3.13 การเชื่อมต่อระหว่าง Arduino A4988 Stepper Driver และ Stepping Motor ชุด B



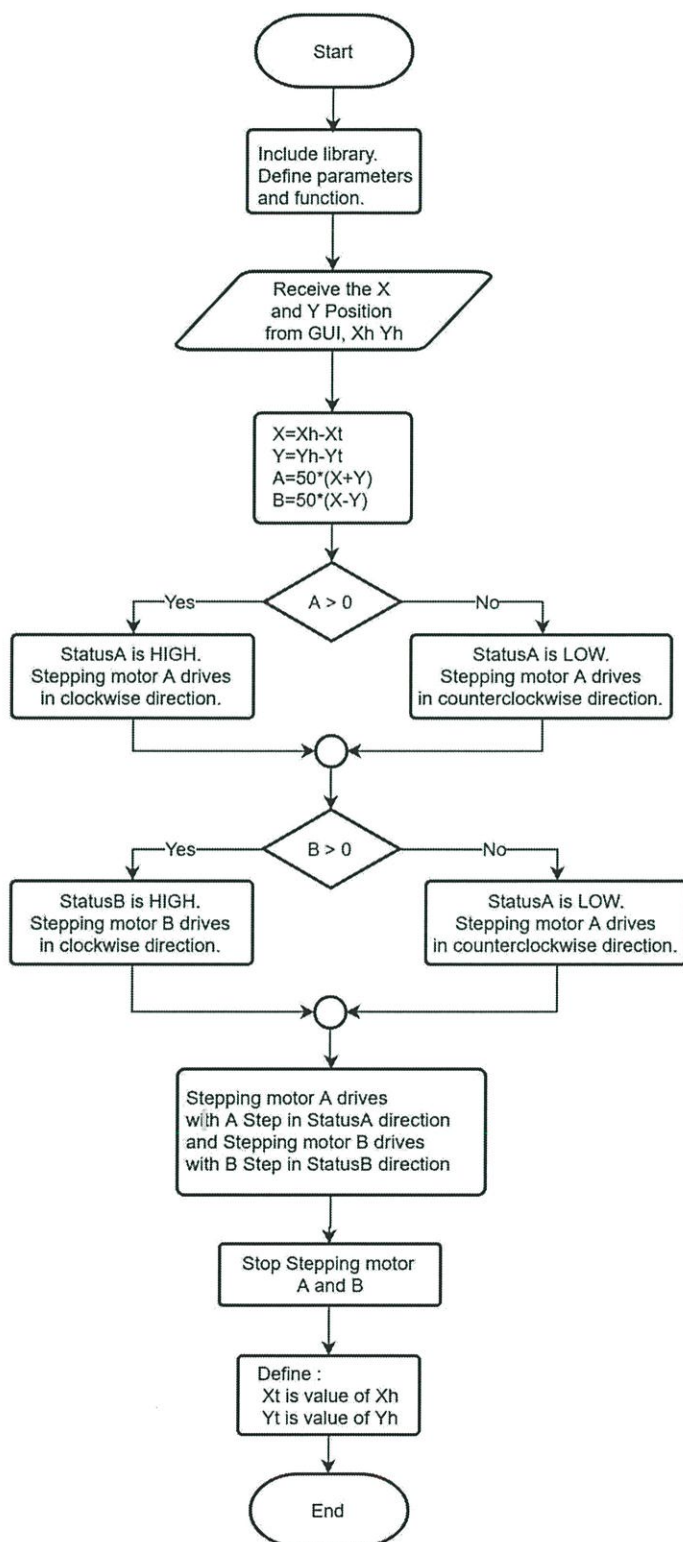
รูปที่ 3.14 รายละเอียดช่องเชื่อมต่อของ Endstop switch [13]

จากรูปที่ 3.14 เป็นรายละเอียดชนิดของพินเชื่อมต่อของ Endstop switch ซึ่งการเชื่อมต่อขาแต่ละขาของ Endstop switch ดังนี้

1. เชื่อมต่อ pin COM ของ Endstop switchA และ Endstop switchB เข้ากับ pin 22 และ 23 ของ Arduino ตามลำดับ
2. เชื่อมต่อ กราวน์ GND ของ Arduino เข้ากับพอร์ต GND ของ Endstop switch

3.1.1.3 ซอฟต์แวร์ควบคุม

ส่วนของโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์ ซึ่งฝังงานการทำงานของชุดเคลื่อนตัวล้อสองแกนประกอบด้วย โปรแกรมเคลื่อนตัวล้อสองแกนเป็นการรับค่าตำแหน่งปลายทางเป็นค่าในพิกัด X และ Y เพื่อเลื่อนตำแหน่งตัวล้อไปยังตำแหน่งนั้นและเก็บค่าพิกัดปัจจุบัน โดยอธิบายดังรูปที่ 3.14 และโปรแกรมตั้งค่าเริ่มต้นของชุดเคลื่อนตัวล้อสองแกนเป็นการนำตัวล้อกลับไปยังตำแหน่งเริ่มต้นของการทำงาน และกำหนดจุดนั้นเป็นตำแหน่ง (0,0) อธิบายดังรูปที่ 3.15



รูปที่ 3.15 ฟังก์ชันการทำงานของโปรแกรมเคลื่อนตัวกลิ้งสองแกน

จากรูปที่ 3.15 การทำงานของไมโครคอนโทรลเลอร์เริ่มต้นจากการกำหนดค่าเริ่มต้นของไมโครคอนโทรลเลอร์ โดยรายละเอียดแสดงดังตารางที่ 3.1

ตารางที่ 3.1 กำหนดค่าเริ่มต้นของไมโครคอนโทรลเลอร์

A	0
B	0
X	0
Y	0
X_h	0
Y_h	0
X_t	0
Y_t	0
StatusA	0
StatusB	0
i	0
j	0

แล้วกำหนดฟังก์ชันการทำงานของมอเตอร์ตัวที่ 1 โดยใช้คำสั่ง `int stepPin1 = 6` และ `int dirPin1 = 5` เป็นการกำหนดการเชื่อมต่อพิน STEP และ DIR ของ A4988 Stepper Driver ตัวที่ 1 เข้ากับพิน 6 และ 5 ของ Arduino กำหนดฟังก์ชันการทำงานของมอเตอร์ตัวที่ 2 โดยใช้คำสั่ง `int stepPin2 = 8` และ `int dirPin2 = 7` เป็นการกำหนดการเชื่อมต่อพิน STEP และ DIR ของ A4988 Stepper Driver ตัวที่ 2 เข้ากับพิน 8 และ 7 ของ Arduino

จากนั้นไมโครคอนโทรลเลอร์จะรับค่าตำแหน่งปลายทางที่ต้องการเลื่อนตัวล้อเป็นพิกัด X และพิกัด Y จาก GUI โดยผ่าน Serial port ซึ่งจะเก็บค่าพิกัดนั้นไว้ในตัวแปร X_h และ Y_h ตามลำดับ และนำค่า X_h และ Y_h ไปเข้าสมการที่ 3.1 และสมการที่ 3.2 ตามลำดับ

$$X = X_h - X_t \quad (3.1)$$

$$Y = Y_h - Y_t \quad (3.2)$$

จากสมการที่ 3.1 และ 3.2 เป็นการหาเวกเตอร์การเคลื่อนที่ของชุดเลื่อนตัวล้อสองแกนจากตำแหน่งเดิมไปยังตำแหน่งใหม่ และนำค่า X และ Y ไปเข้าสมการที่ 3.3 และสมการที่ 3.4 ตามลำดับ

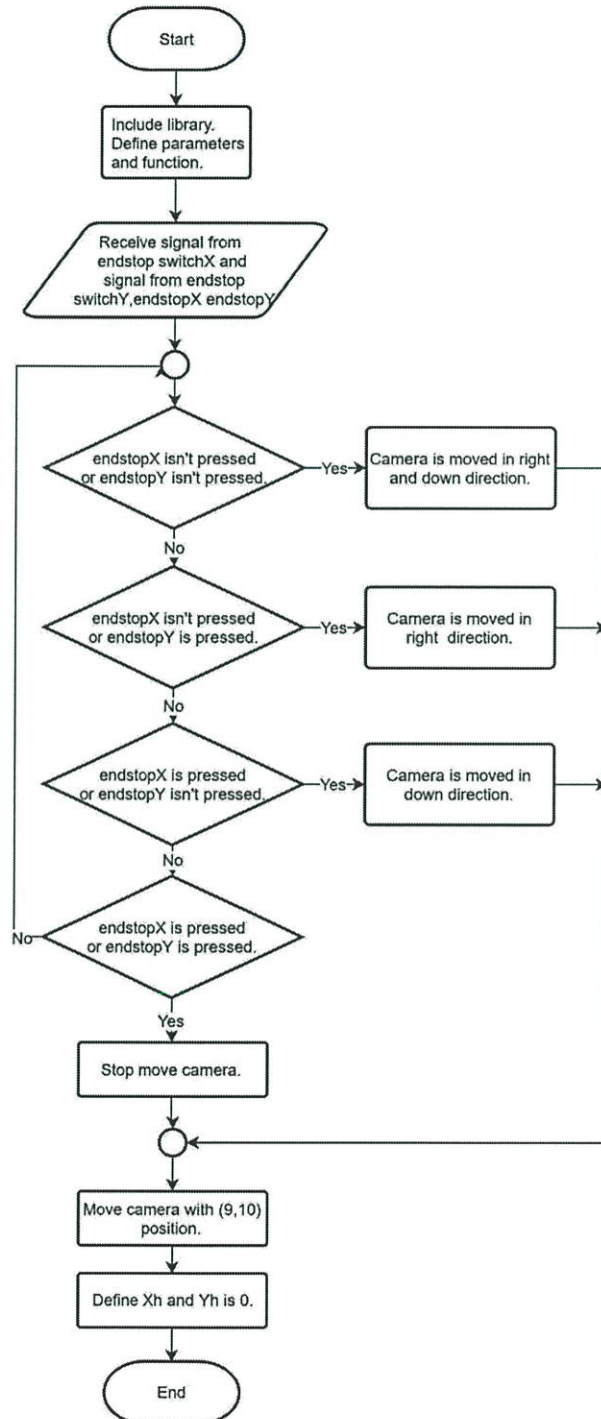
$$A = 50(X + Y) \quad (3.3)$$

$$B = 50(X - Y) \quad (3.4)$$

จากสมการที่ 3.3 และ 3.4 เป็นการหาการหมุนของมอเตอร์แต่ละตัวตามหลักการ การทำงานของชุดเลื่อนตัวล้อสองแกน โดย 50 เป็นค่าที่ทำให้การหมุนของมอเตอร์ทั้งสองตัวอยู่ในหน่วย เซนติเมตรเป็นค่าที่มาจากการทดลอง

จากนั้นตรวจสอบว่าตัวแปร A มีค่ามากกว่า 0 หรือไม่ ถ้าเป็นจริงจะกำหนดให้ตัวแปร StatusA เป็น HIGH ถ้าเป็นเท็จจะกำหนดให้ตัวแปร StatusA เป็น LOW และตรวจสอบว่าตัวแปร B มีค่ามากกว่า 0 หรือไม่ ถ้าเป็นจริงจะกำหนดให้ตัวแปร StatusB เป็น HIGH ถ้าเป็นเท็จจะกำหนดให้ตัวแปร StatusB เป็น LOW แล้วไมโครคอนโทรลเลอร์จะสั่งให้มอเตอร์ตัวที่ 1 ทำงาน โดยการกำหนดการทำงานของ dirPin1 ด้วยตัวแปร StatusA ซึ่งเป็นการกำหนดทิศทางการหมุนของมอเตอร์ตัวที่ 1 ถ้าตัวแปร StatusA เป็น HIGH จะทำให้มอเตอร์ตัวที่ 1 หมุนไปในทิศทางตามเข็มนาฬิกา และถ้าตัวแปร StatusA เป็น LOW จะทำให้มอเตอร์ตัวที่ 1 หมุนไปในทิศทางทวนเข็มนาฬิกา และกำหนดการทำงานของ stepPin1 เป็น HIGH เป็นเวลา 500 μ S แล้วกำหนดเป็น LOW เป็นเวลา 500 μ S เป็นจำนวน A รอบ และสั่งให้มอเตอร์ตัวที่ 2 ทำงาน โดยการกำหนดการทำงานของ dirPin2 ด้วยตัวแปร StatusB ซึ่งเป็นการกำหนดทิศทางการหมุนของมอเตอร์ตัวที่ 2 ถ้าตัวแปร StatusB เป็น HIGH จะทำให้มอเตอร์ตัวที่ 2 หมุนไปในทิศทางตามเข็มนาฬิกา และถ้าตัวแปร StatusB เป็น LOW จะทำให้มอเตอร์ตัวที่ 2 หมุนไปในทิศทางทวนเข็มนาฬิกา และกำหนดการทำงานของ stepPin2 เป็น HIGH เป็นเวลา 500 μ S แล้วกำหนดเป็น LOW เป็นเวลา 500 μ S จำนวน B รอบ ทำให้ตัวล้อเคลื่อนที่ไปยังตำแหน่งที่ผู้ใช้งานต้องการ แล้วมอเตอร์ทั้งสองจะหยุด

ทำงาน และค่าในตัวแปร X_h และ Y_h จะถูกเก็บไว้ในตัวแปร X_i และ Y_i ซึ่งค่าที่ถูกเก็บในตัวแปรทั้งสองคือตำแหน่งปัจจุบันของตัวกล้อง



รูปที่ 3.16 ผังงานการทำงานของโปรแกรมตั้งค่าเริ่มต้นของชุดเคลื่อนตัวกล้องสองแกน

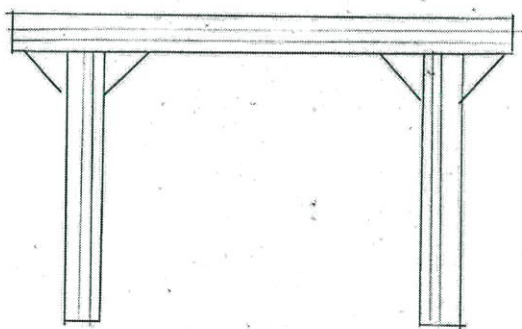
จากรูปที่ 3.16 การทำงานของไมโครคอนโทรลเลอร์เริ่มต้นจากการกำหนดกำหนดค่าเริ่มต้นของไมโครคอนโทรลเลอร์ โดยรายละเอียดแสดงดังตารางที่ 3.1 และกำหนดการทำงานของ Endstop switch ทั้งสองตัวกับพิน 22 และ 23 ของ Arduino โดยใช้คำสั่ง #define endstopY 22 และ #define endstopX 23 แล้วตรวจสอบว่า endstopX และ endstopY ถูกกดหรือไม่ ถ้า endstopX และ endstopY ไม่ถูกกด ไมโครคอนโทรลเลอร์จะสั่งให้ชุดเคลื่อนที่ทั้งสองแกนเคลื่อนที่ลงมาในทิศทางล่าง - ขวา ถ้า endstopX ไม่ถูกกด แต่ endstopY ถูกกด ไมโครคอนโทรลเลอร์จะสั่งให้ชุดเคลื่อนที่ทั้งสองแกนเคลื่อนที่ลงในทิศทางขวามือ ถ้า endstopX ถูกกด แต่ endstopY ไม่ถูกกด ไมโครคอนโทรลเลอร์จะสั่งให้ชุดเคลื่อนที่ทั้งสองแกนเคลื่อนที่ลงในทิศทางล่าง และเมื่อ endstopX และ endstopY ถูกกด ชุดเคลื่อนที่ทั้งสองแกนจะหยุดทำงาน จากนั้นให้ชุดเคลื่อนที่ทั้งสองแกนเคลื่อนที่ด้วยตำแหน่ง (-9,10) และกำหนดให้ตำแหน่งปัจจุบันเป็นตำแหน่งเริ่มต้น (0,0)

3.1.2 การออกแบบเครื่องเปลี่ยนชุดข้อสอบและซอฟต์แวร์ควบคุม

3.1.2.1 โครงสร้าง

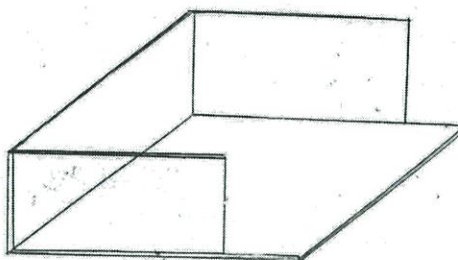
การออกแบบเครื่องเปลี่ยนชุดข้อสอบเป็นการต่อเติมโครงสร้างชุดเคลื่อนที่ตัวทั้งสองแกน โดยประกอบด้วยชุดยกถาดรองข้อสอบ และชุดเปลี่ยนชุดข้อสอบ โดยรายละเอียด และวิธีการประกอบแสดงเป็นลำดับดังต่อไปนี้

1. การประกอบด้วยกถาดรองข้อสอบ โดยนำแท่งอลูมิเนียมโปรไฟล์ยาว 200 mm ทั้งสองมายึดกับแท่งอลูมิเนียมโปรไฟล์ยาว 300 mm ที่ห่างจากขอบทั้งสองด้านเป็นระยะ 5 cm ซึ่งยึดโดยฉากอลูมิเนียม การประกอบด้วยกถาดรองข้อสอบแสดงดังรูปที่ 3.17



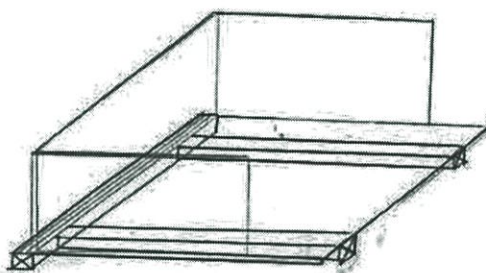
รูปที่ 3.17 การประกอบด้วยกถาดรองข้อสอบ

2. การประกอบถาดรองข้อสอบ โดยนำแผ่นอะคลิลิกรูปทรงสี่เหลี่ยมหนา 2 มิลลิเมตร ขนาด 30x21 เซนติเมตร 1 แผ่น ขนาด 30x10 เซนติเมตร 1 แผ่น และขนาด 15x10 เซนติเมตร 2 แผ่น โดยนำแผ่นอะคลิลิกมาประกอบถาดรองข้อสอบแสดงดังรูปที่ 3.18



รูปที่ 3.18 การประกอบถาดรองข้อสอบ

3. นำถาดรองข้อสอบมายึดกับตัวถาดรองข้อสอบแสดงดังรูปที่ 3.19



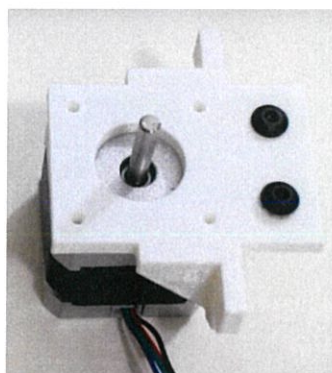
รูปที่ 3.19 การยึดถาดรองข้อสอบกับตัวถาดรองข้อสอบ

3. นำแบริงลูกกลิ้งตรง (linear ball bearing) ที่สวมเพลาดตรง แล้วนำมาติดตั้งกับชิ้นส่วน Z Carriage โดยให้ชิ้นส่วน Z Carriage สวมครอบกับแบริงลูกกลิ้งตรงพร้อมยึดด้วยสกรู และนอต ทำลักษณะนี้ทั้งสองแห่ง การประกอบเพลาดตรงแสดงดังรูปที่ 3.20



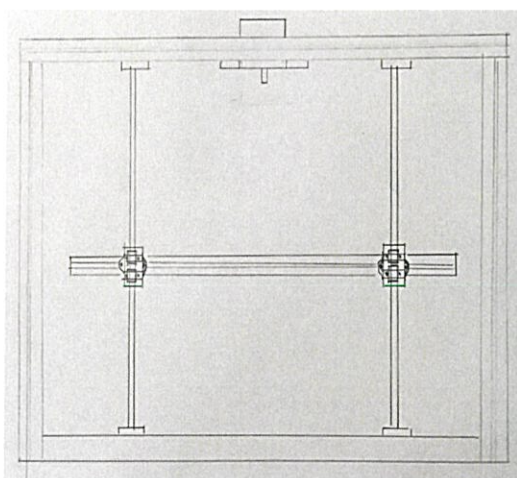
รูปที่ 3.20 การประกอบเพลาดตรง

3. นำชิ้นส่วน Z Motor มาสวมกับ Stepping Motor พร้อมยึดด้วยสกรู และนอต การประกอบ Z Motor แสดงดังรูปที่ 3.21



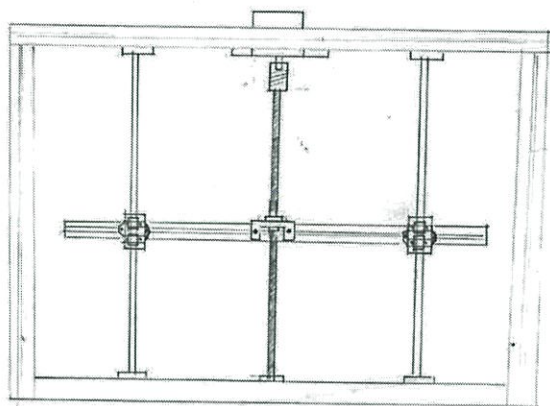
รูปที่ 3.21 การประกอบ Z Motor

4. นำชิ้นส่วน Z Motor ที่สวมกับ Stepping Motor มาติดตั้งกับโครงอลูมิเนียมโปรไฟล์ บริเวณกึ่งกลางระหว่าง Stepping Motor ทั้งสองที่ควบคุมกลไกการทำงานของตัวเลื่อนกลิ้งสองแกนพร้อมยึดด้วยสกรู และนอต นำชิ้นส่วน Z Shaft Clamp มาติดตั้งกับเพลลาตรงที่บริเวณปลายของเพลลาตรงทั้งสองพร้อมยึดด้วยสกรู และนอต จากนั้นนำไปติดตั้งกับโครงอลูมิเนียมโปรไฟล์ โดยติดตั้งให้ห่างจากชิ้นส่วน Z Motor เป็นระยะ 100 mm และนำด้วยฐานรองชุดข้อสอบไปติดตั้งกับชิ้นส่วน Z Shaft Clamp ลักษณะการติดตั้งชิ้นส่วน Z Motor Z Shaft Clamp และฐานรองข้อสอบแสดงดังรูปที่ 3.22



รูปที่ 3.22 ลักษณะการติดตั้งชิ้นส่วน Z Motor Z Shaft Clamp และฐานรองข้อสอบ

5. นำ Coupling ยึดแกนหมุนของ Stepping Motor กับชุด Lead screw แล้วนำชิ้นส่วน Z nut mount มาสวมกับ Copper nut lead และยึดกับตัวฐานรองชุดข้อสอบ ส่วนปลายของอีกด้านของชุด Lead screw ยึดกับโครงอลูมิเนียมโปรไฟล์ โดยใช้ Z Endstop การติดตั้ง ชุด Lead screw แสดงดังรูปที่ 3.23



รูปที่ 3.23 การติดตั้งชุด Lead screw

6. แผ่นอะคลิลิกรูปทรงสี่เหลี่ยมหนา 3 มิลลิเมตร ขนาด 30x30 เซนติเมตร 2 แผ่น

3.1.2.2 การเชื่อมต่อวงจร

การออกแบบวงจรควบคุมการทำงานของเครื่องเปลี่ยนชุดข้อสอบมีอุปกรณ์คือ Arduino Mega 2560 A4988 Stepper Driver Endstop switch Ultrasonics Sensor module และ Stepping Motor โดยการควบคุมชุดยกถาดรองชุดข้อสอบจะใช้ A4988 Stepper Driver และ Stepping Motor ใช้งาน Endstop switch ในฟังก์ชันเลื่อนถาดรองชุดข้อสอบลง เพื่อให้ผู้ใช้งานใส่ชุดข้อสอบในถาดรองชุดข้อสอบ และใช้ Ultrasonics Sensor module ในฟังก์ชันเลื่อนฐานรองชุดข้อสอบขึ้นทำหน้าที่เลื่อนฐานรองชุดข้อสอบขึ้น โดยให้ชุดข้อสอบชุดบนสุดอยู่ในตำแหน่งที่สามารถตรวจจับและรวมคะแนนข้อสอบได้ และมีรายละเอียด และขั้นตอนการเชื่อมต่ออุปกรณ์ดังนี้

จากตารางที่ 3.13 เป็นรายละเอียดชนิดของช่องเชื่อมต่อของ A4988 Stepper Driver ซึ่งการเชื่อมต่อขาแต่ละขาของ A4988 Stepper Driver ดังนี้

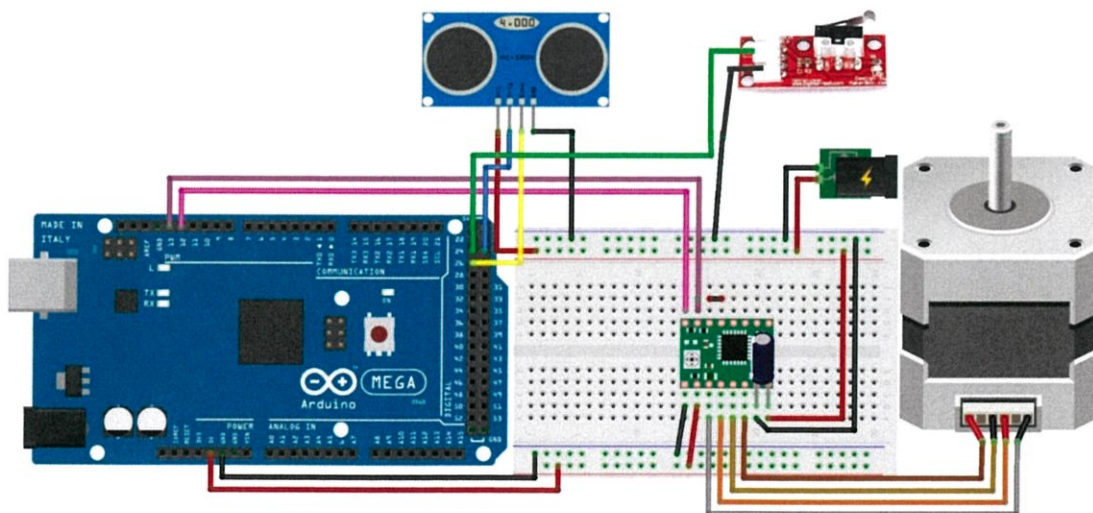
8. เชื่อมต่อพิน STEP และ DIR ของ A4988 Stepper Driver เข้ากับพิน 13 และ 12 ของ Arduino ตามลำดับ
9. เชื่อมต่อพิน 1B 1A 2A และ 2B ของ A4988 Stepper Driver เข้ากับสายที่ 1 2 3 และ 4 ของ Stepping Motor ตามลำดับ
10. เชื่อมต่อพิน RESET และ SLEEP ของ A4988 Stepper Driver เข้าด้วยกัน
11. เชื่อมต่อพิน 5V ของ Arduino เข้ากับพิน VCC ของ A4988 Stepper Driver

12. นำตัวเก็บประจุขนาด 100 μ F เชื่อมต่อขาบวก และขาลบเข้ากับพิน VMOT และ GND ของ A4988 Stepper Driver ตามลำดับ
13. เชื่อมต่อไฟเลี้ยง +12V เข้ากับพิน VMOT ของ A4988 Stepper Driver
14. เชื่อมต่อกราวด์ GND ของ Arduino และ GND ของไฟเลี้ยงเข้ากับพอร์ต power GND

จากตารางที่ 3.17 และ 3.18 มีการเชื่อมต่อขาแต่ละขาของ Ultrasonics Sensor module และ Endstop switch ดังนี้

3. เชื่อมต่อ pin Trig และ Echo ของ Ultrasonics Sensor module เข้ากับ pin 25 และ 26 ของ Arduino ตามลำดับ
4. เชื่อมต่อ pin COM ของ Endstop switch เข้ากับ pin 24 ของ Arduino
5. เชื่อมต่อ pin 5V ของ Arduino เข้ากับพอร์ต VCC ของ Ultrasonics Sensor module
6. เชื่อมต่อ กราวด์ GND ของ Arduino เข้ากับพอร์ต GND ของ Ultrasonics Sensor module และ Endstop switch

การเชื่อมต่อวงจรของชุดยกถาดรองชุดข้อสอบแสดงดังรูปที่ 3.24

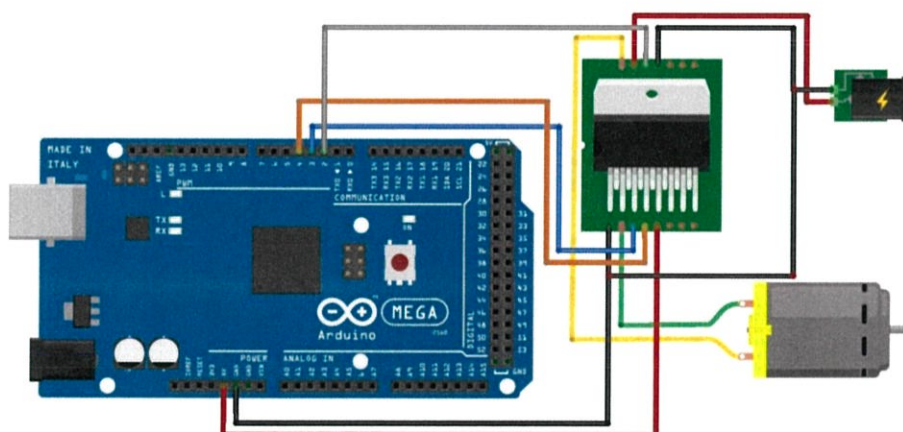


รูปที่ 3.24 ตัวอย่างการเชื่อมต่อสายระหว่าง Arduino A4988 Stepper Driver Stepping motor Ultrasonics Sensor module และ Endstop switch

การต่อสายไฟในอุปกรณ์ระหว่าง Arduino IC L298N และ DC motor รายละเอียดของเชื่อมต่อของ IC L298N แสดงดังรูปที่ 3.25

จากตารางที่ 3.14 มีการเชื่อมต่อขาแต่ละขาของ Arduino IC L298N และ DC motor ดังนี้

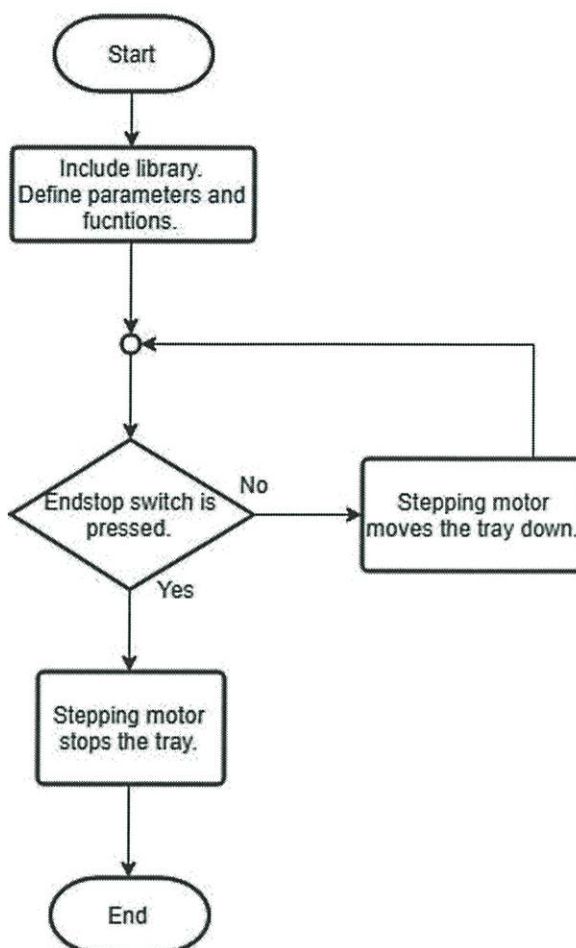
1. เชื่อมต่อ pin 6 ENA 5 IN1 และ 7 IN2 ของ IC L298N เข้ากับ pin 2 3 และ 4 ของ Arduino ตามลำดับ
2. เชื่อมต่อ pin 2 OUT1 และ 3 OUT2 ของ IC L298N เข้ากับ DC motor
3. เชื่อมต่อ pin 5V ของ Arduino เข้ากับพอร์ต 9 VSS ของ IC L298N
4. เชื่อมต่อไฟเลี้ยง 12 V เข้ากับพอร์ต 4 VS ของ IC L298N
5. เชื่อมต่อ กราวด์ GND ของ Arduino เข้ากับพอร์ต 1 CSA 8 GND ของ IC L298N



รูปที่ 3.25 ตัวอย่างการเชื่อมต่อสายระหว่าง Arduino IC L298N และ DC motor

3.1.2.3 ซอฟต์แวร์ควบคุม

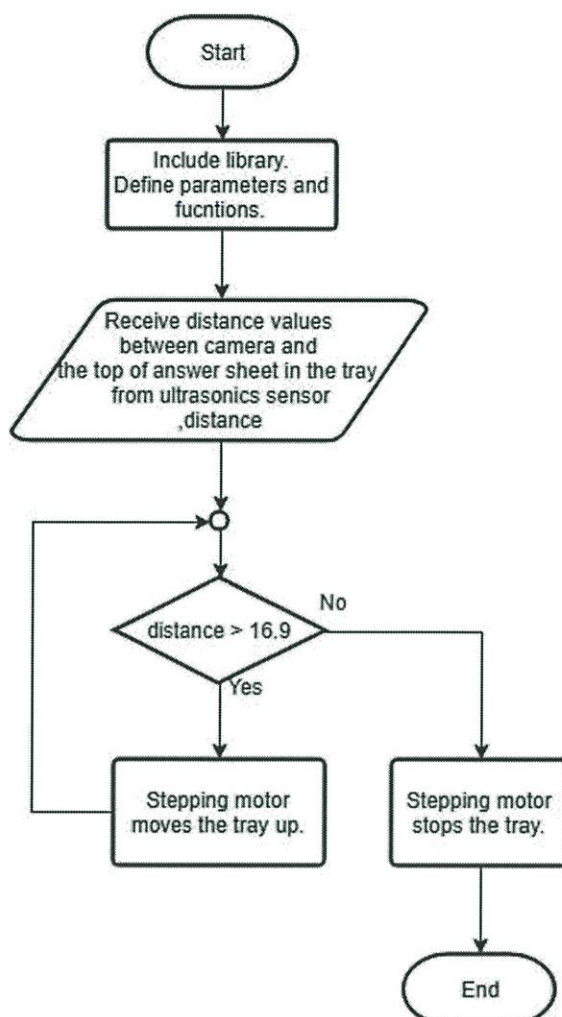
ส่วนของโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์ ซึ่งฝังงานการทำงานของชุดยกลาดรองชุดข้อสอบประกอบด้วย ฟังก์ชันเลื่อนลาดรองชุดข้อสอบลงเป็นการเลื่อนฐานรองชุดข้อสอบลงเพื่อให้ผู้ใช้งานนำชุดกระดาษคำตอบใส่ลงในฐานรองชุดข้อสอบ ฝังงานการทำงานของฟังก์ชันเลื่อนลาดรองชุดข้อสอบลงอธิบายดังรูปที่ 3.26 ฟังก์ชันเลื่อนลาดรองชุดข้อสอบขึ้นเป็นการเลื่อนลาดรองชุดข้อสอบขึ้น เพื่อให้ชุดข้อสอบชุดบนสุดอยู่ในตำแหน่งที่พร้อมจะรวมคะแนน ฝังงานการทำงานของฟังก์ชันเลื่อนลาดรองชุดข้อสอบขึ้นอธิบายดังรูปที่ 3.27 และฟังก์ชันเปลี่ยนชุดข้อสอบโดยเลื่อนชุดข้อสอบชุดบนสุดออกจากฐานรองชุดข้อสอบเพื่อรองรับการทำงานในชุดต่อไป ฝังงานการทำงานของฟังก์ชันเปลี่ยนชุดข้อสอบอธิบายดังรูปที่ 3.26



รูปที่ 3.26 ผังงานการทำงานของฟังก์ชันเลื่อนฐานรองชุดข้อสอบลง

จากรูปที่ 3.26 การทำงานของฟังก์ชันเลื่อนฐานรองชุดข้อสอบลงเริ่มต้นจากการกำหนดตัวแปร กำหนดการใช้งาน Stepping Motor โดยใช้คำสั่ง `int stepPin3 = 13` และ `int dirPin3 = 12` เป็นการกำหนดการเชื่อมต่อพิน STEP และ DIR ของ A4988 Stepper Driver เข้ากับพิน 13 และ 12 ของ Arduino กำหนดการทำงานของ Endstop switch กับพิน 24 ของ Arduino โดยใช้คำสั่ง `#define endstopZ 24` และกำหนดการใช้งาน Ultrasonics Sensor module คือ `trigPin` และ `echoPin` เข้ากับพิน 25 และ 26 ของ Arduino โดยใช้คำสั่ง `int trigPin = 25` และ `echoPin = 26`

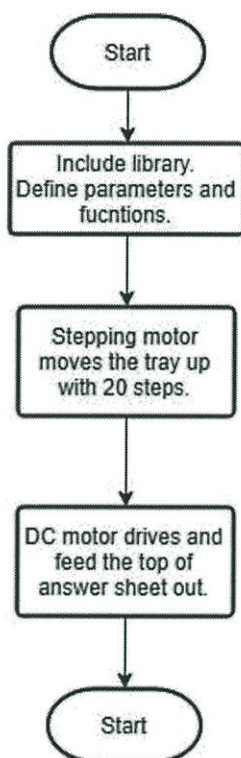
จากนั้นตรวจสอบการทำงานของ Endstop switch ว่าถูกกดหรือไม่ ถ้า Endstop switch ยังไม่ถูกกด จะทำให้ Stepping Motor หมุนสลับทิศทางเข็มนาฬิกาเพื่อให้ชุดยกฐานรองชุดข้อสอบเคลื่อนที่ลง และหยุดหมุนเมื่อฐานรองชุดข้อสอบเลื่อนลงมาจนกด Endstop switch แสดงว่าฐานรองชุดข้อสอบเคลื่อนที่ลงมายังระยะต่ำสุด และพร้อมให้ผู้ใช้วางใส่ชุดข้อสอบลงไปยังฐานรองชุดข้อสอบ



รูปที่ 3.27 ผังงานการทำงานของฟังก์ชันเลื่อนฐานรองชุดข้อสอบขึ้น

จากรูปที่ 3.27 การทำงานของฟังก์ชันเลื่อนฐานรองชุดข้อสอบขึ้นเริ่มต้นจากการทำงานของไมโครคอนโทรลเลอร์นำเข้าไลบรารีเพิ่มเติม โดยใช้คำสั่ง `#include <Stepper.h>` เพื่อให้สามารถใช้คำสั่งเพิ่มเติมเพื่อควบคุม Stepping Motor กำหนดค่าตัวแปรต่างๆ กำหนดค่า `stepsInRev = 200`; คือ Step ทั้งหมดใน 1 รอบการหมุนโดยคำนวณจากนำ 360° มาหารด้วยค่า Step Angle ของ Stepping Motor จากตารางที่ 3.15 ซึ่งมีค่าเป็น 1.8° กำหนดตัวแปร `stepsInRev` ให้มีค่าเป็น 200 กำหนดตัวแปร `num_of_step` ให้มีค่าเป็น 1 กำหนด pin สำหรับควบคุม Stepping Motor โดยใช้คำสั่ง `Stepper myStepper1(stepsInRev, 9, 10, 11, 12)`; กำหนด pin สำหรับการใช้งาน Ultrasonics Sensor module คือ `trigPin` เป็น 25 และ `echoPin` เป็น 26 แล้วกำหนดคุณสมบัติการทำงานให้อุปกรณ์ต่างๆ ซึ่งคำสั่งทั้งหมดจะบรรจุไว้ภายใต้ฟังก์ชัน `setup()`

จากนั้นรับค่าระยะทางระหว่างชุดข้อสอบในฐานรองชุดข้อสอบชุดบนสุด กับ Ultrasonics Sensor module ที่ติดอยู่กับตัวกล่อง เก็บไว้ในตัวแปร distance แล้วตรวจสอบว่าตัวแปร distance มีค่ามากกว่า 16.9 หรือไม่ ซึ่งค่า 16.9 เป็นระยะที่กล่องตรวจจับ และรวมคะแนนข้อสอบ ถ้าตัวแปร distance มีค่ามากกว่า 16.9 แสดงว่าชุดข้อสอบในฐานรองชุดข้อสอบชุดบนสุดอยู่ในระยะที่ต่ำกว่าระยะที่กล่องตรวจจับ จะทำให้ Stepping Motor หมุนสแต็ปทีละตามเข็มนาฬิกาเพื่อให้ชุดยกฐานรองชุดข้อสอบเคลื่อนที่ขึ้น เมื่อฐานรองชุดข้อสอบเคลื่อนที่ขึ้นจะทำให้ตัวแปร distance มีค่าลดลง และจะหยุดเมื่อแปร distance มีค่าเป็น 16.9 แสดงว่าชุดข้อสอบในฐานรองชุดข้อสอบชุดบนสุดอยู่ในระยะที่กล่องตรวจจับ และพร้อมรวมคะแนนข้อสอบ



รูปที่ 3.28 ผังงานการทำงานของเปลี่ยนชุดข้อสอบ

จากรูปที่ 3.28 การทำงานของฟังก์ชันเปลี่ยนชุดข้อสอบเริ่มต้นจากการทำงานของไมโครคอนโทรลเลอร์กำหนด pin สำหรับควบคุม DC Motor คือ pwm เป็น 2 กำหนด in_1 เป็น 3 และกำหนด in_2 เป็น 4 กำหนดตัวแปร var2 ให้มีค่าเป็น และ paper ให้มีค่าเป็น 1 แล้วกำหนดคุณสมบัติการทำงานให้อุปกรณ์ต่างๆ ซึ่งคำสั่งทั้งหมดจะบรรจุไว้ภายใต้ฟังก์ชัน setup()

จากนั้นให้ Stepping Motor หมุนไป 20 สเต็ปทิศตามเข็มนาฬิกาเพื่อให้ชุดยกฐานรองชุดข้อสอบเคลื่อนที่ขึ้นเป็นระยะเท่ากับความหนาของชุดข้อสอบ 1 ชุด แล้วให้ DC Motor หมุนเพื่อดึงชุดข้อสอบชุดบนสุดออก

3.1.3 การออกแบบโปรแกรมประมวลผลภาพ

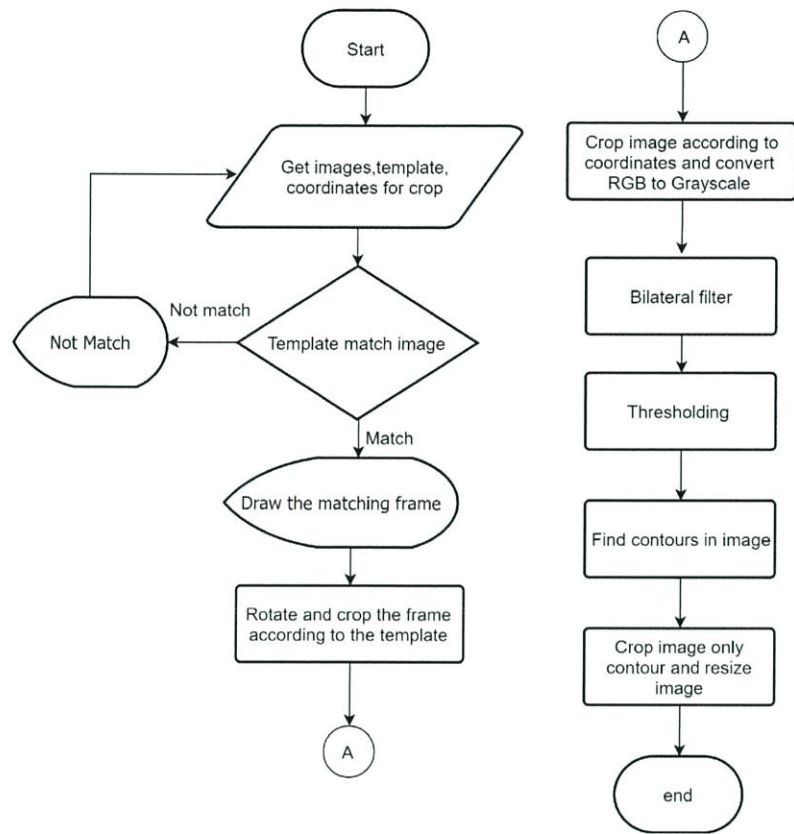
การออกแบบโปรแกรมประมวลผลภาพสามารถแบ่งการทำงานได้ 2 ฟังก์ชันและสามารถอธิบายขั้นตอนการทำงานได้ดังนี้

3.1.3.1 ฟังก์ชันตัดกรอบตัวเลข

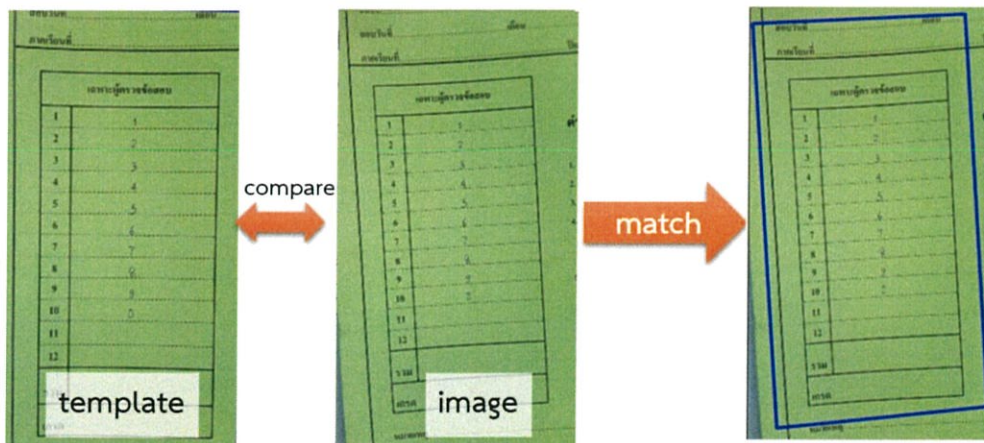
ฟังก์ชันตัดกรอบตัวเลขต้องมีการลดสัญญาณรบกวนภายในภาพและปรับแก้ปัญหาของภาพจากกล้องเช่น กระจกคาชค่าตอบเอียง สีของกระจกคาชค่าตอบ การตรวจสอบความถูกต้องของกระจกคาชค่าตอบ เป็นต้น ซึ่งกระบวนการทำงานของโปรแกรมสามารถอธิบายได้ด้วยแผนผังการทำงานดังรูปที่ 3.29 ซึ่งสามารถลำดับการทำงานของโปรแกรมสามารถอธิบายได้ดังนี้

1. โปรแกรมจะรับภาพจากกล้อง (image) ภาพต้นแบบ (template) และตำแหน่งพิกัดสำหรับตัดกรอบช่องกรอกคะแนน (coordinates for crop)

2. โปรแกรมจะเริ่มตรวจจับสอบภาพจากกล้อง โดยการเปรียบเทียบภาพจากกล้องกับภาพต้นแบบ (template match image) หากภาพจากกล้องตรงกับภาพต้นแบบโปรแกรมจะตีกรอบบริเวณที่ภาพจากกล้องตรงกับภาพต้นแบบ (draw the matching frame) ได้ดังรูปที่ 3.30 และหากภาพจากกล้องไม่ตรงกับภาพต้นแบบโปรแกรมจะแสดงข้อความ “Not match” แสดงบนจอภาพแสดงได้ดังรูปที่ 3.31 และกลับไปรับภาพใหม่จากกล้อง



รูปที่ 3.29 แผนผังการทำงานของฟังก์ชันตัดกรอบตัวเลข

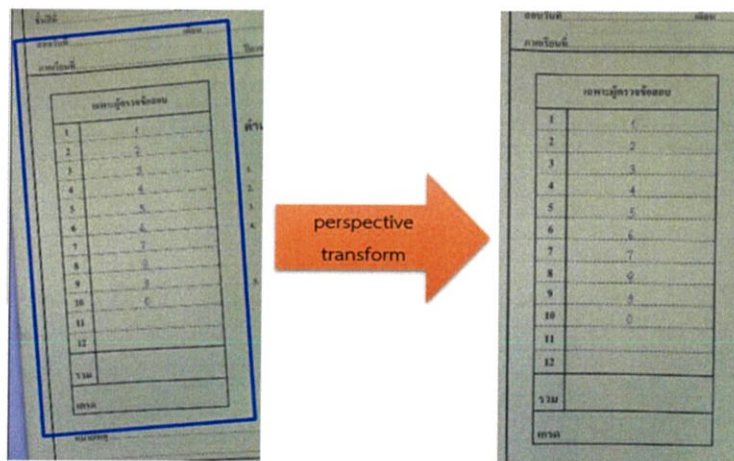


รูปที่ 3.30 ผลของโปรแกรมเมื่อภาพจากกล้องตรงกับภาพต้นแบบ



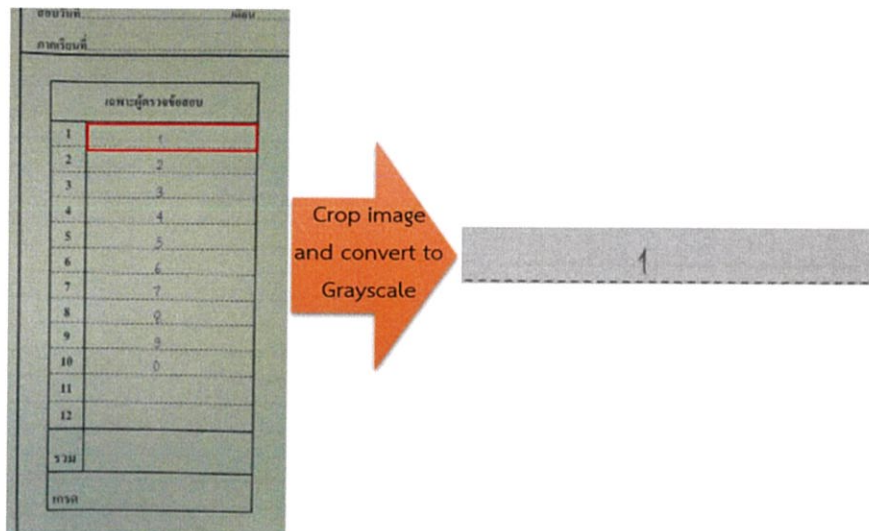
รูปที่ 3.31 ผลของโปรแกรมเมื่อภาพจากกล้องไม่ตรงกับภาพต้นแบบ

3. โปรแกรมจะนำภาพที่ได้เข้าสู่กระบวนการแก้ไขความเอียงและตัดกรอบเฉพาะส่วนที่เหมือนกับภาพต้นแบบ (Rotate and crop the frame according to the template) ด้วยกระบวนการ การแปลงมุมมองของภาพ แสดงผลการทำงานได้ดังรูปที่ 3.32



รูปที่ 3.32 ผลของกระบวนการ perspective transform

4. โปรแกรมจะตัดกรอบช่องกรอกคะแนนโดยใช้พิกัดสำหรับตัดกรอบ (coordinates for crop) และแปลงภาพจากภาพสี (RGB) เป็นภาพขาวเทา (Grayscale) ได้ดังรูปที่ 3.33



รูปที่ 3.33 ผลของการตัดกรอบช่องกรอกคะแนนและแปลงสีภาพเป็นภาพขาวเทา (Grayscale)

5. โปรแกรมจะนำภาพที่ถูกตัดกรอบช่องกรอกคะแนนแล้วเข้าสู่ Bilateral filter โดยกำหนดค่าขนาดของตัวกรอง ค่า sigma color และ ค่า sigma space เพื่อกำจัดสัญญาณรบกวนภายในภาพได้ดังรูปที่ 3.34



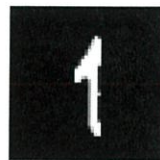
รูปที่ 3.34 ภาพหลังจากเข้าสู่ Bilateral filter

6. ภาพที่ผ่านตัวกรองแล้วจะเข้าสู่กระบวนการ Thresholding เพื่อสร้างความแตกต่างระหว่างพื้นหลังกับตัวเลขให้มากขึ้น โดยกำหนดค่า Threshold โดยหากเกินค่า Threshold พิกเซลนั้นจะถูกแทนด้วย 0 และหากต่ำกว่าจะถูกแทนด้วย 1 ดังนั้นจะได้ผลลัพธ์เป็นภาพขาวดำแสดงได้ดังรูปที่ 3.35



รูปที่ 3.35 ภาพหลังการกระบวนการ Thresholding

7. หลังจากการทำ Thresholding ภาพจะเข้าสู่การ Find contours โดยวิธี Moor Neighbor เป็นวิธีที่ช่วยให้การสแกนหา contours ได้เร็วขึ้น ซึ่งเป็นกระบวนการหาความแตกต่างของพิกเซล และบันทึกตำแหน่งที่เกิดความแตกต่างระหว่างพิกเซลทำให้สามารถระบุตำแหน่งของตัวเลขได้ เนื่องจากบริเวณที่เป็นตัวเลขเป็นสีขาวและพื้นหลังเป็นสีดำ และนำค่าตำแหน่งที่ได้มาตัดกรอบเฉพาะตัวเลข ซึ่งสามารถแสดงผลได้ดังรูปที่ 3.36 และบันทึกค่าตำแหน่งพิกัดแรกที่พบ contours

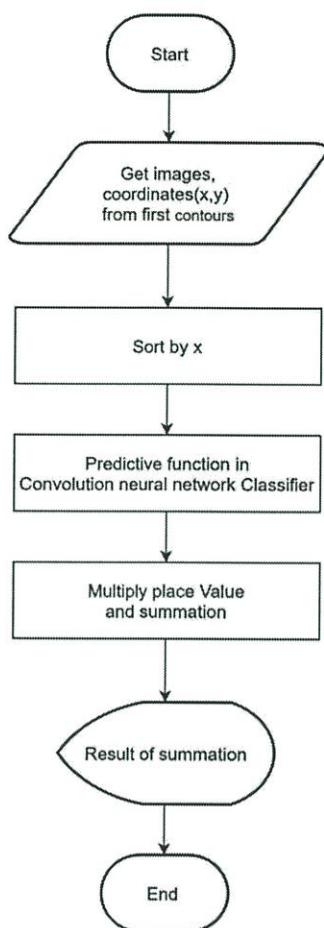


รูปที่ 3.36 ภาพหลังจากการ Find contours

8. ปรับขนาดภาพให้มีขนาดเท่ากับ 48×48 พิกเซล ให้เท่ากับอินพุตที่ถูกกำหนดไว้ในโปรแกรมจำแนกตัวเลขด้วยกระบวนการเรียนรู้และจดจำแบบ CNN เมื่อจบฟังก์ชันจะได้เอาต์พุตเป็นภาพไบนารีขนาด 48×48 และพิกัดแรกที่พบ contours ของแต่ละตัวเลข

3.1.3.1 ฟังก์ชันลำดับค่าประจำหลัก

ฟังก์ชันลำดับค่าประจำหลักเป็นฟังก์ชันที่คูณค่าประจำหลักของตัวเลขโดยเรียงจากตำแหน่งการเขียนตัวเลข โดยเป็นฟังก์ชันที่รับค่าต่อจากฟังก์ชันตัดกรอบตัวเลข ซึ่งสามารถแสดงการทำงานได้ดังแผนผังดังรูปที่ 3.37

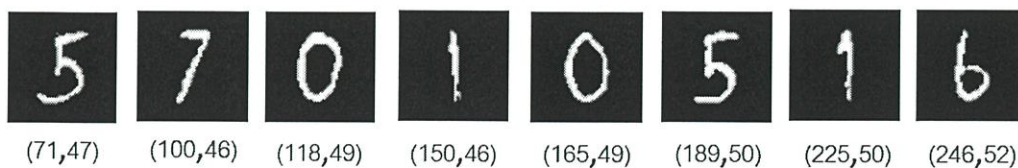


รูปที่ 3.37 แผนผังการทำงานของฟังก์ชันลำดับค่าประจำหลัก

จากรูปที่ 3.37 สามารถลำดับการทำงานของโปรแกรมสามารถอธิบายได้ดังนี้

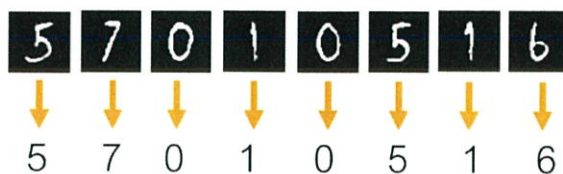
1. รับภาพจากฟังก์ชันตัดกรอบตัวเลขและค่าตำแหน่งแรกที่พบคอนทัวร์
2. นำข้อมูลตำแหน่งที่พบคอนทัวร์มาเรียงลำดับค่าพิกัด x โดยเรียงจากน้อยไปมาก แสดง

ได้ดังรูปที่ 3.38



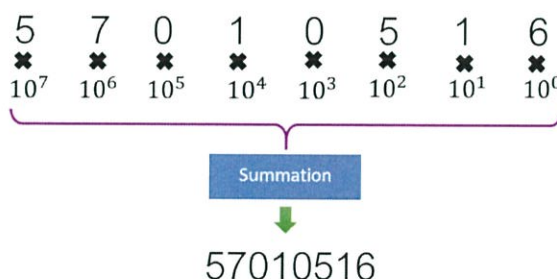
รูปที่ 3.38 ผลการเรียงลำดับจากค่าพิกัด x

3. นำภาพที่ได้เข้าสู่โปรแกรมจำแนกตัวเลขจากลายมือด้วย convolution neural network จะได้เอาต์พุตเป็นตัวในช่วงตั้งแต่ 0 ถึง 9 จากแต่ละภาพ แสดงได้ดังรูปที่ 3.39



รูปที่ 3.39 ผลการจำแนกตัวเลขจากลายมือ

3. หลังจากผลการเรียงจะทำให้ทราบจำนวนตัวเลขทั้งหมดที่มี และนำไปคูณกับค่าประจำหลักในระบบเลขฐาน 10 ในแต่ละตัวเลข แสดงได้ดังรูปที่ 3.40



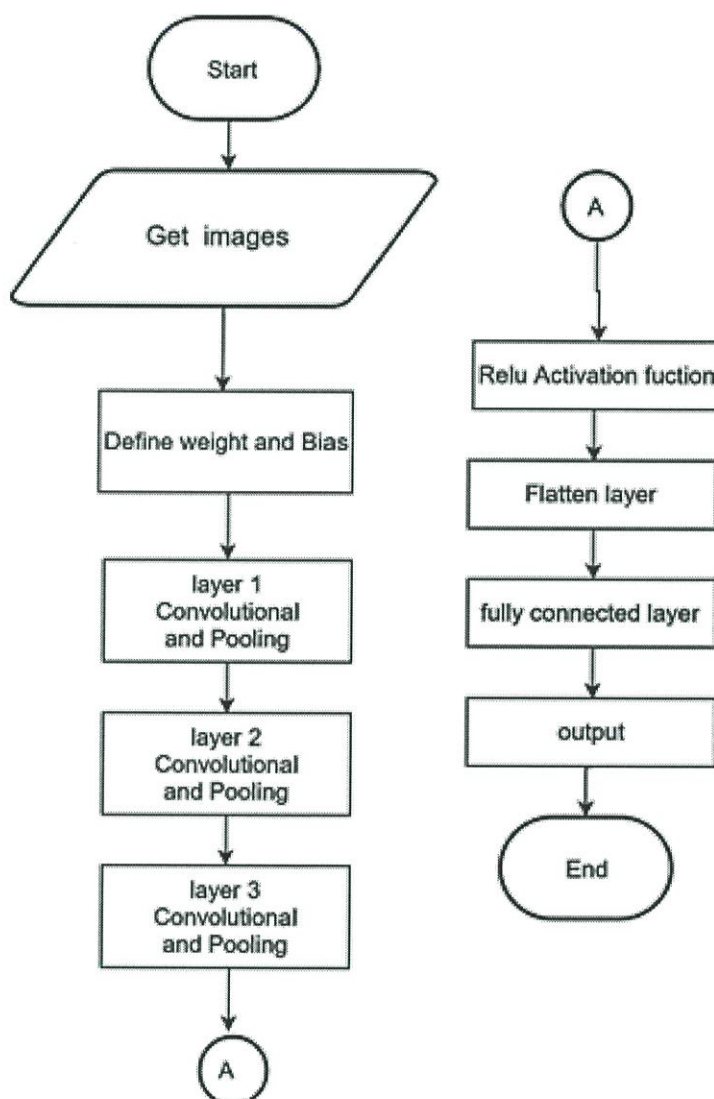
รูปที่ 3.40 การทำงานในลำดับที่ 3

4. นำผลรวมที่ได้จากการคูณเลขประจำหลักเป็นผลการจำแนกตัวเลขในบริเวณที่ถูกตัดกรอบเป็นการสิ้นสุดการทำงาน

3.1.4 การออกแบบโปรแกรมจำแนกตัวเลขด้วยกระบวนการเรียนรู้และจดจำแบบ CNN

โปรแกรมจำแนกตัวเลขด้วยกระบวนการเรียนรู้และจดจำแบบ CNN เป็นการลำดับโครงสร้างการทำงานของ CNN โดยออกแบบให้ภาพอินพุตของโปรแกรมมีขนาด 48×48 พิกเซล เพราะเป็นขนาดที่ไม่เล็กจนทำให้ภาพตัวเลขลายมือสูญเสียรูปแบบ และเป็นขนาดที่ไม่ใหญ่จนทำให้การทำงานของโปรแกรมล้าช้า กำหนดชั้นการหาคุณลักษณะจากภาพ (feature extraction) เท่ากับ 3 ชั้น ซึ่งประกอบด้วยการทำ convolution pooling และ Relu activation function หลังจากนั้นเข้า

ฟังก์ชันจะเข้าสู่การจำแนก (classification) จากคุณลักษณะนั้น ๆ ในชั้น fullyconnected ซึ่งสามารถอธิบายลำดับการทำงานได้ดังรูปที่ 3.41



รูปที่ 3.41 ลำดับการทำงานของฟังก์ชันโครงสร้างโมเดล

จากรูปที่ 3.41 มีการทำงานในแต่ละลำดับคือ

1. รับภาพจากฟังก์ชันตัดกรอบตัวเลขซึ่งเป็นภาพไบนารีและมีขนาดเท่ากับ 48×48 พิกเซล
2. กำหนดจำนวนตัวกรองถ่วงน้ำหนัก ขนาดของตัวกรองถ่วงน้ำหนัก จำนวนตัวถ่วงน้ำหนักในชั้น fully connected และไบอัสในแต่ละชั้นจะถูกกำหนดตามจำนวนของตัวถ่วงน้ำหนัก

3. ชั้นที่ 1 ฟังก์ชันจะนำตัวกรองถ่วงน้ำหนักทั้งหมด convolution กับภาพอินพุตและบวกด้วยไบอัสของแต่ละตัวกรอง จากนั้นนำภาพทั้งหมดที่ได้เข้าสู่กระบวนการ pooling โดยกำหนดค่า Pooling size เท่ากับ 2×2 และ Stride เท่ากับ 2 เพื่อลดขนาดลง 2 เท่า
4. ชั้นที่ 2 และ 3 จะรับภาพจากชั้นก่อนหน้า และทำเช่นเดียวกับข้อที่ 3
5. เออต์พุตของชั้นที่ 3 จะเข้าสู่ Relu activation function ซึ่งจะให้ออต์พุตมีเฉพาะค่าที่เป็นบวกเท่านั้น
6. เข้าสู่ชั้น Flatten เป็นการเปลี่ยนมิติอาร์เรย์จาก 2 มิติเป็น 1 มิติ
7. เข้าสู่ชั้น fully connected เพื่อนำคุณลักษณะต่าง ๆ เข้าสู่การจำแนก (classification) ซึ่งผลลัพธ์ของชั้นนี้จะได้เป็นอาร์เรย์ขนาด 1×10

3.1.5 การออกแบบโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้

โปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ในปริญญาโทฉบับนี้ เป็นการรับภาพจากตัวกล้องของเครื่องเปลี่ยนชุดข้อสอบอัตโนมัติมาเข้าสู่กระบวนการประมวลผลภาพในการจำแนกตัวเลขลายมือ ผู้ใช้สามารถสั่งการทำงานผ่านหน้าต่าง GUI ได้โดยสามารถสั่งการทำงานได้ดังนี้ 1.การควบคุมตัวกล้อง ผู้ใช้สามารถควบคุมตัวกล้องเลื่อนไปยังตำแหน่งที่ต้องการตรวจจับตัวเลขภายในชุดข้อสอบ เพื่อให้สามารถรองรับการใช้ชุดข้อสอบรูปแบบอื่น ๆ ได้ 2.การป้อนชุดข้อสอบออกจากเครื่องเปลี่ยนชุดข้อสอบ ผู้ใช้สามารถป้อนชุดข้อสอบออก หลังจากผ่านการตรวจจับตัวเลขลายมือแล้ว 3.การควบคุมถาดรองชุดข้อสอบ ผู้ใช้สามารถควบคุมถาดรองชุดข้อสอบให้เลื่อนลงมาเพื่อให้สามารถใส่ชุดข้อสอบเข้าไปในเครื่องเปลี่ยนชุดข้อสอบได้

ออกแบบหน้าต่าง GUI โดยมีทั้งหมด 6 หน้า แสดงแผนผังการทำงานของโปรแกรม GUI คร่าว ๆ ได้ดังรูปที่ 3.42 ซึ่งประกอบด้วย หน้าต่างเริ่มต้น (Start Page) หน้าต่างเลื่อนตัวกล้อง (Move Camera Page) หน้าต่างสร้างกรอบตรวจจับ (Draw Rectangle Page) หน้าต่างยืนยันการลากกรอบ (Confirm Draw Page) หน้าต่างยืนยันชุดข้อสอบ (Confirm Answersheet Page) และ หน้าต่างตรวจจับตัวเลข (Detect Page)

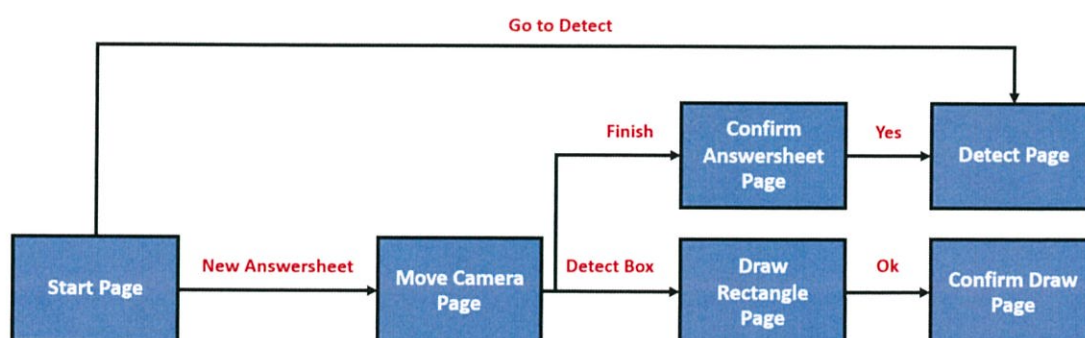
1. หน้าต่างเริ่มต้น เป็นหน้าที่ผู้ใช้สามารถเลือกได้ว่าต้องการตรวจจับตัวเลขหรือตั้งค่าชุดข้อสอบ
2. หน้าต่างเลื่อนตัวกล้อง เป็นหน้าที่ GUI เชื่อมต่อการทำงานกับโปรแกรมเลื่อนตัวกล้อง ผู้ใช้สามารถเลื่อนตัวกล้องไปยังตำแหน่งที่ต้องการได้

3. หน้าต่างสร้างกรอบตรวจจับ เป็นหน้าที่ผู้ใช้สามารถลากกรอบสี่เหลี่ยมภายในภาพ จากตัวกล้อง

4. หน้าต่างยืนยันการลากกรอบ เป็นหน้าที่ผู้ยืนยันว่าการลากกรอบตรวจจับนั้น ถูกต้อง ผู้ใช้สามารถตั้งชื่อได้ว่ากรอบที่ลากคืออะไร

5. หน้าต่างยืนยันชุดข้อสอบ เป็นหน้าที่ผู้ยืนยันชื่อชุดข้อสอบ โดยผู้ใช้สามารถตั้งชื่อได้เอง

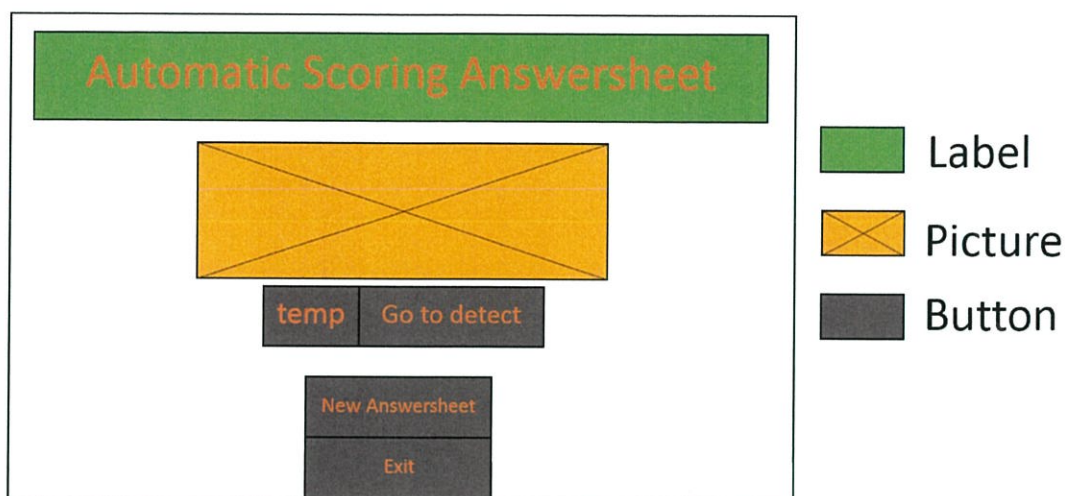
6. หน้าต่างตรวจจับตัวเลขเป็นหน้าที่ผู้ใช้สามารถตรวจจับ จำแนกตัวเลขลายมือ ควบคุมไมโครคอลโทรลเลอร์ป้อนชุดข้อสอบออกจากเครื่อง ควบคุมถาดรองชุดข้อสอบในการเลื่อน ขึ้นและลง เพื่อให้ผู้ใช้สามารถนำชุดข้อสอบใส่ในถาดรองชุดข้อสอบได้



รูปที่ 3.42 แผนผังการทำงานของโปรแกรม GUI

3.1.5.1 หน้าต่างเริ่มต้น

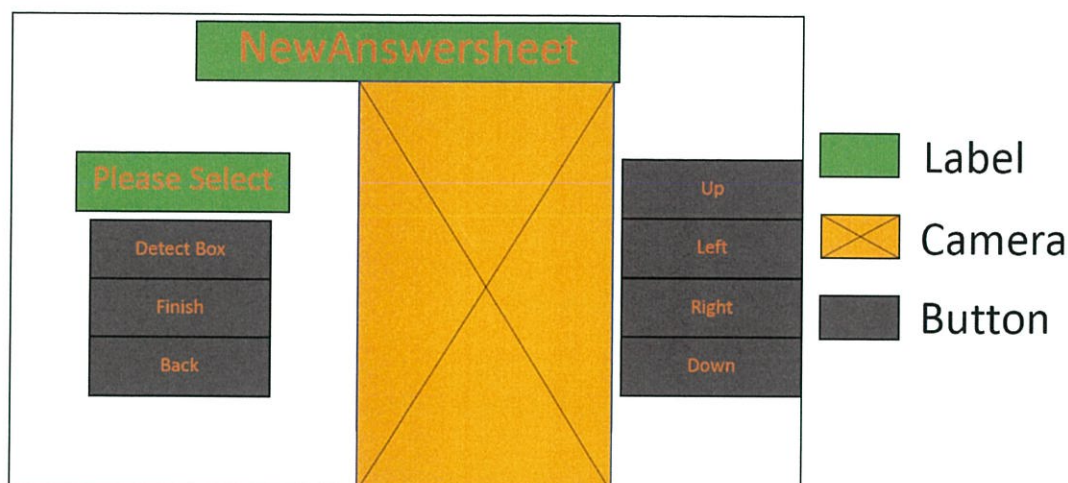
เป็นหน้าต่างที่ผู้ใช้สามารถเลือกที่ไปยังหน้าต่างตรวจจับตัวเลข และหน้าต่างเลื่อนตัว กล้อง โดยภายในหน้าต่างเริ่มต้นประกอบด้วย label “Automatic Scoring Answersheet” ภาพ ตราสัญลักษณ์สจล. ปุ่ม “temp” ที่ให้ผู้ใช้เลือกชุดข้อสอบที่ผ่านการตั้งค่าลากกรอบตรวจจับตัวเลข ปุ่ม “Go to detect” เพื่อไปยังหน้าต่างตรวจจับตัวเลข ปุ่ม “New Answersheet” เพื่อไปยัง หน้าต่างเลื่อนตัวกล้อง และปุ่ม “Exit” เพื่อออกจากโปรแกรม แสดงภาพจำลองหน้าต่างได้ดังรูปที่ 3.43



รูปที่ 3.43 ภาพจำลองหน้าต่างเริ่มต้น

3.1.5.2 หน้าต่างเลื่อนตัวกล้อง

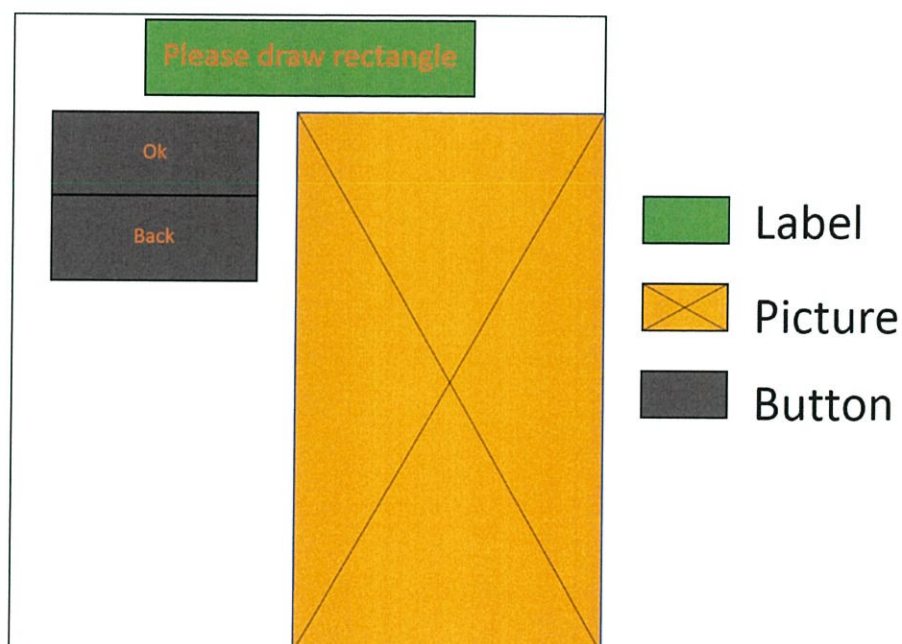
เป็นหน้าต่างที่ผู้ใช้สามารถควบคุมชุดเลื่อนตัวกล้องสองแกนไปยังตำแหน่งที่ต้องการ ลากกรอบตรวจจับตัวเลข เนื่องจากภาพที่ได้จากตัวกล้องไม่สามารถเห็นได้ทั่วชุดข้อสอบ จึงต้องมีการควบคุมชุดเลื่อนตัวกล้องสองแกน โดยแรกเริ่มโปรแกรมจะส่งค่าเลข 5 ไปยังโปรแกรมควบคุมชุดเลื่อนตัวกล้อง เพื่อให้ชุดเลื่อนตัวกล้องเลื่อนมายังตำแหน่งกึ่งกลางของตารางชุดข้อสอบ ภายในหน้าต่างประกอบด้วย label “New Answersheet” label “Please Select” ปุ่ม “Detect Box” เพื่อไปยังหน้าต่างลากกรอบตรวจจับตัวเลข ปุ่ม “Finish” เพื่อให้ผู้ใช้ไปยังหน้าต่างยืนยันชื่อชุดข้อสอบ ปุ่ม “Back” เพื่อย้อนกลับไปยังหน้าต่างเริ่มต้น ภาพจากตัวกล้อง ปุ่ม “Up” “Left” “Right” “Down” เพื่อสั่งให้ชุดเลื่อนตัวกล้องเลื่อนขึ้น ซ้าย ขวา และลงตามลำดับ ซึ่งเมื่อผู้ใช้กดปุ่ม “Up” โปรแกรมจะส่งค่าเลข 1 ไปยังโปรแกรมควบคุมชุดเลื่อนตัวกล้อง เพื่อให้ชุดเลื่อนตัวกล้องเลื่อนขึ้น กดปุ่ม “Left” โปรแกรมจะส่งค่าเลข 2 เพื่อเลื่อนซ้าย กดปุ่ม “Right” โปรแกรมจะส่งค่าเลข 3 เพื่อเลื่อนขวา และกดปุ่ม “Down” โปรแกรมจะส่งค่าเลข 4 เพื่อเลื่อนลง แสดงภาพจำลองหน้าต่างได้ดังรูปที่ 3.44



รูปที่ 3.44 ภาพจำลองหน้าจอเลื่อนตัวกล้อง

3.1.5.3 หน้าต่างลากกรอบตรวจจับตัวเลข

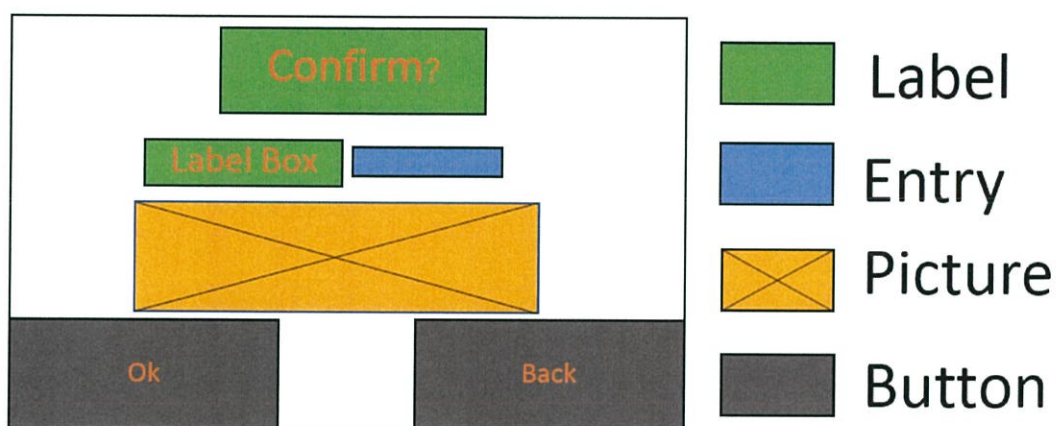
เป็นหน้าต่างที่ผู้ใช้สามารถลากกรอบภายในภาพจากตัวกล้อง ภายในหน้าต่างประกอบด้วย label “Please draw rectangle” ปุ่ม “Ok” เพื่อไปยังหน้าต่างยืนยันการลากกรอบ โปรแกรมจะตัดภาพและเปลี่ยนสัดส่วนของภาพมาเป็นขนาดจริง ปุ่ม “Back” เพื่อย้อนกลับไปยังหน้าต่างเลื่อนตัวกล้อง และภาพจากตัวกล้อง ที่มีอัตราส่วนของภาพน้อยกว่าขนาดจริงครึ่งหนึ่ง เนื่องจากถ้าใช้อัตราส่วนขนาดจริงจะทำให้ไม่สามารถเห็นหน้าต่างนี้ได้ครบ แสดงภาพจำลองหน้าต่างได้ดังรูปที่ 3.45



รูปที่ 3.45 ภาพจำลองหน้าต่างเลื่อนตัวกล้อง

3.1.5.4 หน้าต่างยืนยันการลากกรอบ

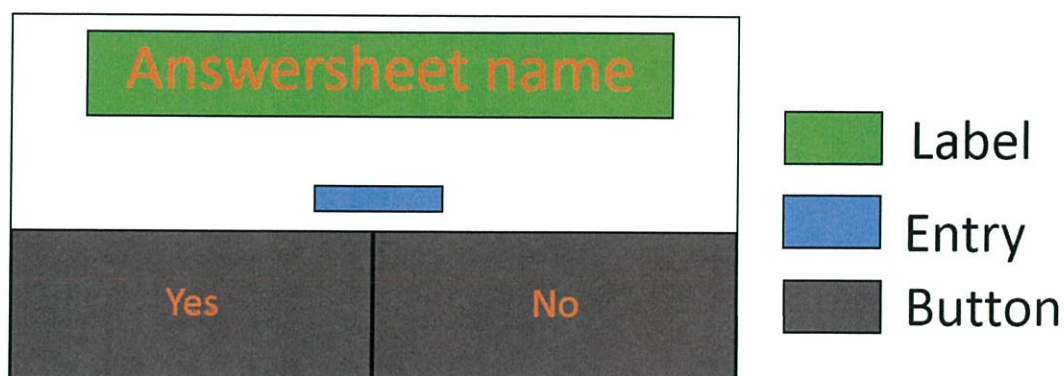
เป็นหน้าต่างที่ผู้ใช้ต้องยืนยันว่าภาพที่ลากกรอบคืออะไรตามที่ผู้ใช้สามารถกำหนดได้เอง ภายในหน้าต่างประกอบด้วย label “Confirm?” label “Label Box” ช่องว่างสำหรับตั้งชื่อภาพว่าคืออะไร ภาพที่ตัดมาจากการลากกรอบโดยจะเปลี่ยนอัตราส่วนมาเป็นขนาดจริง ปุ่ม “Ok” เพื่อยืนยันและกลับไปยังหน้าต่างลากกรอบตรวจจับ และปุ่ม “Back” เพื่อกลับไปยังหน้าต่างลากกรอบตรวจจับ แสดงภาพจำลองหน้าต่างได้ดังรูปที่ 3.46



รูปที่ 3.46 ภาพจำลองหน้าต่างยืนยันการลากกรอบ

3.1.5.5 หน้าต่างยืนยันชื่อชุดข้อสอบ

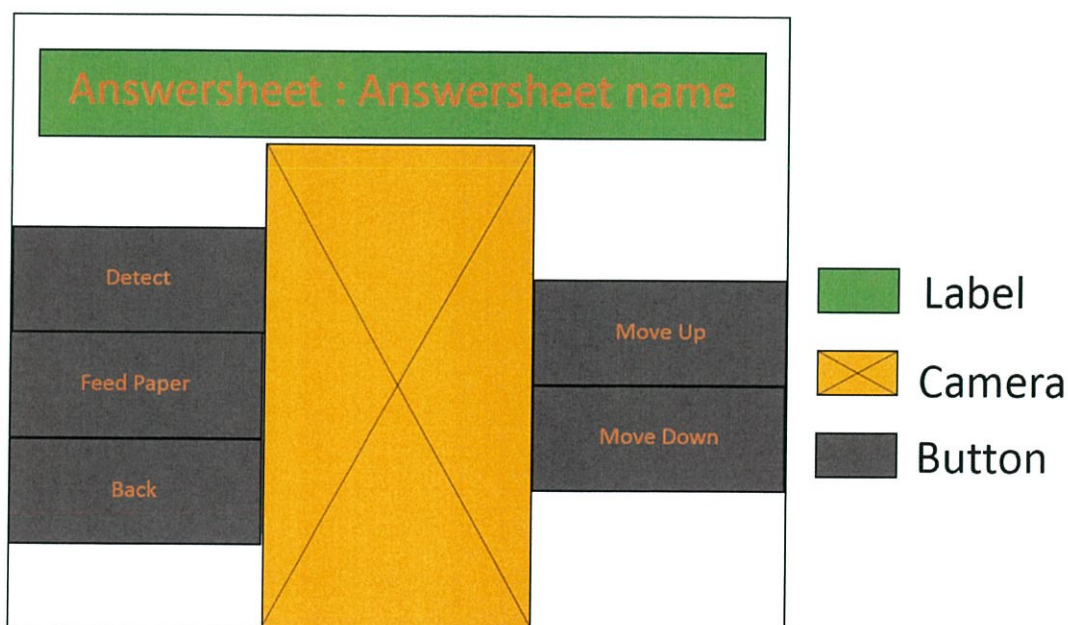
ผู้ใช้ต้องยืนยันว่าชุดข้อสอบที่ผ่านการลากกรอบตรวจจับคืออะไรตามที่ผู้ใช้สามารถกำหนดได้เอง ภายในหน้าต่างประกอบด้วย label “Answersheet name” ช่องว่างสำหรับตั้งชื่อชุดข้อสอบ ปุ่ม “Yes” เพื่อยืนยัน และปุ่ม “Back” เพื่อกลับไปยังหน้าต่างเลื่อนตัวกล้อง แสดงภาพจำลองหน้าต่างได้ดังรูปที่ 3.47



รูปที่ 3.47 ภาพจำลองหน้าต่างยืนยันชื่อชุดข้อสอบ

3.1.5.6 หน้าต่างตรวจจับตัวเลข

เป็นหน้าต่างที่ผู้ใช้สามารถตรวจจับ จำแนกตัวเลขลายมือ เปลี่ยนชุดข้อสอบออกจากเครื่อง ควบคุมมาตรฐานชุดข้อสอบในการเลื่อนขึ้นและลง การเปลี่ยนชุดข้อสอบออกจากเครื่อง โปรแกรมจะส่งค่า “0” ไปยังโปรแกรมป้อนชุดข้อสอบ เพื่อป้อนชุดข้อสอบออก การควบคุมมาตรฐานชุดข้อสอบโปรแกรมจะส่งค่า “9” ไปยังโปรแกรมควบคุมมาตรฐานชุดข้อสอบเพื่อให้มาตรฐานชุดข้อสอบเลื่อนขึ้นจนพอดีกับลูกยางป้อนกระดาษ ส่งค่า “8” ไปยังโปรแกรมควบคุมมาตรฐานชุดข้อสอบเพื่อให้มาตรฐานชุดข้อสอบเลื่อนลง ทำให้ผู้ใช้สามารถนำชุดข้อสอบใส่ในมาตรฐานชุดข้อสอบได้ ภายในหน้าต่างประกอบด้วย label ที่เป็นชื่อชุดข้อสอบที่ผู้ใช้ตั้งชื่อไว้แล้ว ภาพจากตัวกล้อง ปุ่ม “Detect” เพื่อทำการจำแนกตัวเลขลายมือ ปุ่ม “Feed Paper” เพื่อป้อนชุดข้อสอบออก ปุ่ม “Back” เพื่อกลับไปยังหน้าต่างเริ่มต้น ปุ่ม “Move Up” เพื่อเลื่อนมาตรฐานชุดข้อสอบขึ้น ปุ่ม “Move Down” เพื่อมาตรฐานชุดข้อสอบลง โดยการทำงานของหน้าต่างตรวจจับตัวเลขเริ่มจากโปรแกรมส่งค่า “5” ไปยังโปรแกรมเลื่อนตัวกล้อง เพื่อเลื่อนตัวกล้องมาที่ตำแหน่งกึ่งกลางของมาตรฐานชุดข้อสอบ จากนั้นโปรแกรมจะส่งค่า “7” ไปยังโปรแกรมเลื่อนตัวกล้องเพื่อเตรียมรับค่าตำแหน่งที่ตัวกล้องจะเลื่อนไปยังตำแหน่งที่ผู้ใช้ได้ตั้งค่าไว้แล้ว โปรแกรมจะส่งค่าตำแหน่งพิกัด x และ y ไปยังโปรแกรมเลื่อนตัวกล้องตามลำดับ ตัวกล้องจะเลื่อนไปยังตำแหน่ง x และ y หลังจากนั้นโปรแกรมจะสามารถตรวจจับตัวเลขโดยการกดปุ่ม “Detect” จากผู้ใช้ แสดงภาพจำลองหน้าต่างได้ดังรูปที่ 3.48



รูปที่ 3.48 ภาพจำลองหน้าต่างตรวจจับตัวเลข

3.2 เครื่องมือที่ใช้ในการทดลอง


3.2.1 ส่วนเครื่องเปลี่ยนชุดข้อสอบ

เครื่องมือที่ใช้ในการทดลองแบ่งออกเป็น 2 ส่วนคือ เครื่องมือการออกแบบพร้อมประกอบโครงสร้างชุดเลื่อนตัวกลิ้งสองแกน และเครื่องมือการออกแบบวงจรควบคุมการทำงานของชุดเลื่อนตัวกลิ้งสองแกน

3.1.2.1 เครื่องมือการออกแบบพร้อมประกอบโครงสร้างชุดเลื่อนตัวกลิ้งสองแกน

1.) อลูมิเนียมโปรไฟล์ (Aluminum Profile)

ตารางที่ 3.2 อุปกรณ์อลูมิเนียมโปรไฟล์

	ขนาด	จำนวน
	450 mm	4
	400 mm	4
	350 mm	4
	300 mm	1
	200 mm	2

2.) แผ่นอะคลิลิก

ตารางที่ 3.3 อุปกรณ์แผ่นอะคลิลิก

	ขนาด	จำนวน
	30x30 cm	2
	30x21 cm	1
	30x10 cm	1
	15x10 cm	2


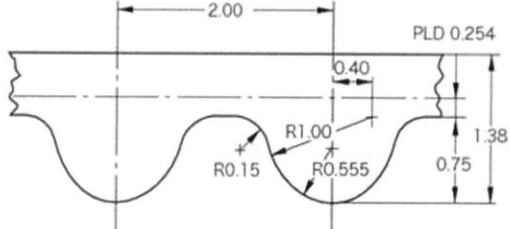
3.) ฉากอลูมิเนียม และชุดนอตยึดร่อง

ตารางที่ 3.4 อุปกรณ์ฉากอลูมิเนียม และชุดนอตยึดร่อง

อุปกรณ์	รูปภาพ	จำนวน
ฉากอลูมิเนียม		28 ตัว
ชุดนอตยึดร่อง		80 ชุด

4.) สายพาน

ตารางที่ 3.5 อุปกรณ์สายพาน

	ความกว้าง	6 mm
	ความยาว	1.8 m
	จำนวน	2 เส้น
		

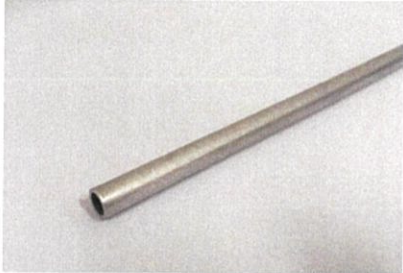
5.) เพลาตรง (linear shaft)

ตารางที่ 3.6 อุปกรณ์เพลาตรง (linear shaft)

	ขนาดหน้าตัด	8 mm
	ความยาว	400 mm
	จำนวน	4 แท่ง


6.) ท่อตรง (linear tube)

ตารางที่ 3.7 อุปกรณ์ท่อตรง (linear tube)

	ขนาดหน้าตัด	8 mm
	ความยาว	350 mm
	จำนวน	4 แท่ง


7.) แบริ่งลูกกลิ้งตรง (linear ball bearing)

ตารางที่ 3.8 อุปกรณ์แบริ่งลูกตรง (linear ball bearing)

	เส้นรอบวงภายใน	8 mm
	เส้นรอบวงภายนอก	15 mm
	จำนวน	8 ตัว

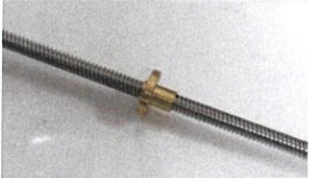
8.) ลูกปืนตลับขนาดเล็ก (small ball bearing)

ตารางที่ 3.9 อุปกรณ์แบริ่งลูกตรง (linear ball bearing)

	เส้นรอบวงภายใน	4 mm
	เส้นรอบวงภายนอก	8 mm
	ความหนา	5 mm
	จำนวน	10 ตัว


9.) ชุด Lead screw

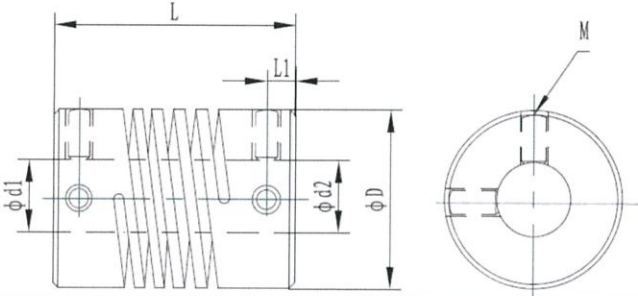
ตารางที่ 3.10 อุปกรณ์ชุด Lead screw

	ขนาดหน้าตัด	8 mm
	ความยาว	400 mm

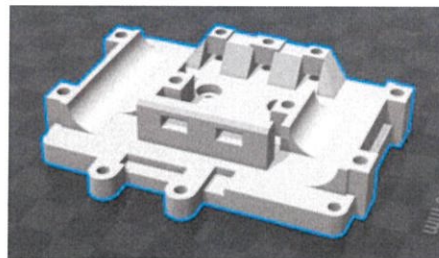
10.) Coupling

ตารางที่ 3.11 อุปกรณ์ Coupling

	เส้นรอบวงภายใน d1	5 mm
	เส้นรอบวงภายใน d2	8 mm
	เส้นรอบวงภายนอก D	15 mm
	ความยาว	25 mm

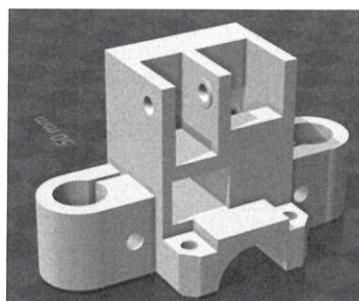


11.) X Carriage



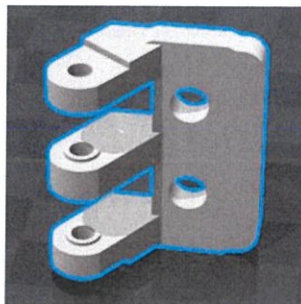
รูปที่ 3.49 ชิ้นส่วน X Carriage

12.) XY Joiner จำนวน 2 ชั้น



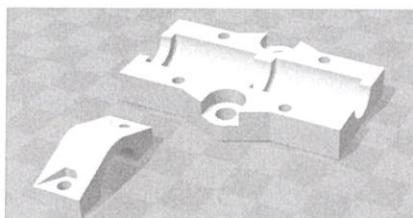
รูปที่ 3.50 ชิ้นส่วน XY Joiner

13.) XY Idler จำนวน 2 ชิ้น



รูปที่ 3.51 ชิ้นส่วน XY Idler

14.) Z Carriage จำนวน 2 ชิ้น



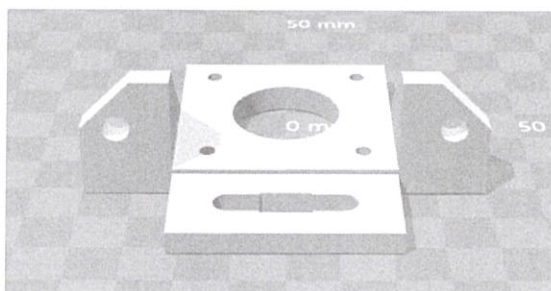
รูปที่ 3.52 ชิ้นส่วน X Carriage

15.) Z nut mount



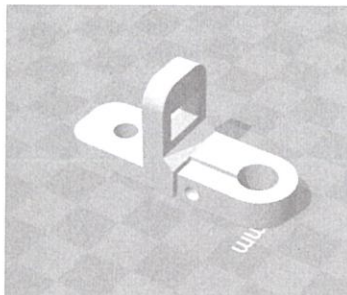
รูปที่ 3.53 ชิ้นส่วน Z nut mount

16.) Z Motor



รูปที่ 3.54 ชิ้นส่วน Z Motor

17.) Z Shaft Clamp จำนวน 4 ชิ้น



รูปที่ 3.55 ชิ้นส่วน Z Shaft Clamp

3.1.2.2 เครื่องมือการออกแบบวงจรควบคุมการทำงานของชุดเคลื่อนตัวล้อสองแกน

1) Arduino Mega 2560

ตารางที่ 3.12 อุปกรณ์ Arduino Mega 2560 [14]

	Microcontroller	ATmega2560
	Operating Voltage	5V
	Digital I/O Pins	54 (of which 15 provide PWM output)
	Analog Input Pins	16
	Clock Speed	16 MHz


2) A4988 Stepper Driver

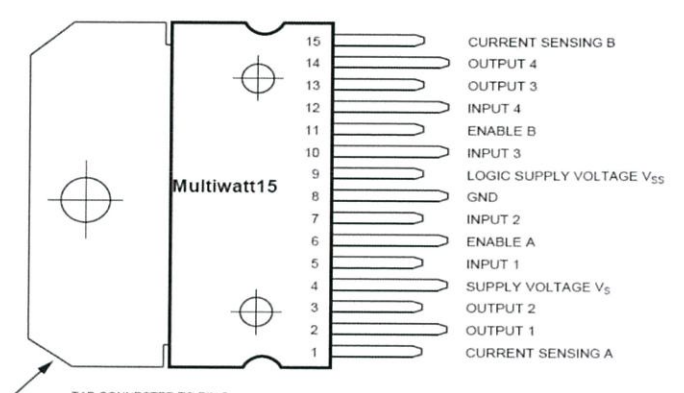
ตารางที่ 3.13 อุปกรณ์ A4988 Stepper Driver [12]

<p> ENABLE MS1 MS2 MS3 RESET SLEEP STEP DIR </p> <p> VMOT GND 2B 2A 1A 1B VDD GND </p>	Operating Voltage	8 - 35 V
	Logic Voltage	3 - 5.5 V
	Continuous current per phase	1 A
	Maximum current per phase	2 A
	Microstep resolutions	Full, 1/2, 1/4, 1/8, and 1/16

3.) IC L298N


ตารางที่ 3.14 อุปกรณ์ IC L298N [15]

	Operating Voltage	UP TO 46 V
	Logic Voltage	4.5-7V
	Current - Output	2A
	Voltage - Load	4.8 V ~ 46 V
	Package / Case	15-Multiwatt




4.) Stepping Motor จำนวน 3 ตัว

ตารางที่ 3.15 อุปกรณ์ Stepping Motor [16]

	Step Angle	1.8°
	Voltage	2.55 V
	Rated Current	1.8 A
	Diameter of axle	5 mm
	Static torque	4.2 kg.cm
	จำนวน	3 ตัว

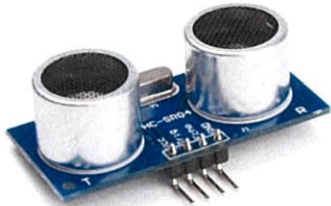
5.) DC Motor

ตารางที่ 3.16 อุปกรณ์ DC Motor [2]

	Model	ZGA25RP
	Voltage	6~24 V
	Output Power	0.5~2 W
	Rotation Rate	100 RPM


6.) Ultrasonics Sensor module

ตารางที่ 3.17 อุปกรณ์ Ultrasonics Sensor module [5]

	Working voltage	DC 5 V
	Working temperature	0 ~ + 70
	Induction Angle	Less than 15
	Detection range	2 cm to 4 m
	Detecting precision	0.3 cm + 1%

7.) Endstop switch

ตารางที่ 3.18 อุปกรณ์ Endstop switch [13]

 <p> — COM — GND — VCC </p>	Working voltage	3.3-5 V
	จำนวน	3 ตัว
	connection VCC (ramps +)	red line
	connected to GND (ramps -)	Black line
	connection SIGNAL (ramps s)	green line

3.2.2 ส่วนโปรแกรมประมวลผลภาพ โปรแกรมจำแนกตัวเลขและโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้

3.3.2.1 คอมพิวเตอร์ ASUS Transformer Book Flip TP550LD



รูปที่ 3.56 คอมพิวเตอร์ ASUS Transformer Book Flip TP550LD [17]

ตารางที่ 3.19 คุณสมบัติของคอมพิวเตอร์ ASUS Transformer Book Flip TP550LD

Processor	Intel® Core™ i5 4200U Processor
Operation System	Windows 10
Memory	Onboard Memory 4 GB
Graphic	NVIDIA® Geforce® 820M with 2GB DDR3
Optical Drive	Super-Multi DVD (On selected models)
Card Reader	2 in 1 card reader (SD/SDHC)
Camera	VGA Web Camera
Networking	Integrated 802.11 b/g/n 10/100/1000/Gigabits Base T BT 4.0 support (on WLAN+ BT 4.0 combo card)
Interface	1 x Combo audio jack 1 x USB 3.0 port(s) 2 x USB 2.0 port(s) 1 x RJ45 LAN Jack for LAN insert 1 x HDMI 1 x SD card reader

ตารางที่ 3.19 คุณสมบัติของคอมพิวเตอร์ ASUS Transformer Book Flip TP550LD (ต่อ)

Audio	Built-in Speakers and Microphone SonicMaster
Battery	2 Cells 38 Whrs Polymer Battery
Power Adapter	เอาต์พุต : 19 V DC, 3.42 A, 65W อินพุต : 110 – 240 V AC, 50/60 Hz universal

3.2.2.2 Web Camera oker FULL HD Auto Focus HD-386



รูปที่ 3.57 Web Camera OKER FULL HD Auto Focus HD-386 [18]

ตารางที่ 3.20 คุณสมบัติของ Web Camera oker FULL HD Auto Focus HD-386

Resolution	2.0 megapixel
Frame rate	1920 x 1080 per 30 fps
Interface	USB 2.0
Lens quality	High
Focus length	30 mm - Infinity

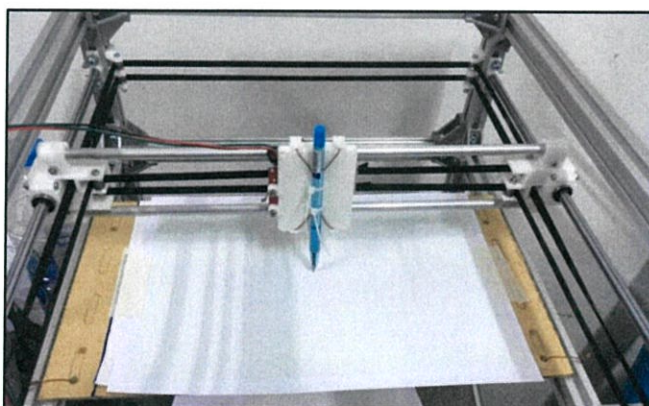
3.3 การจัดเก็บผลการทดลอง

3.3.1 การประกอบเครื่องเปลี่ยนชุดข้อสอบ

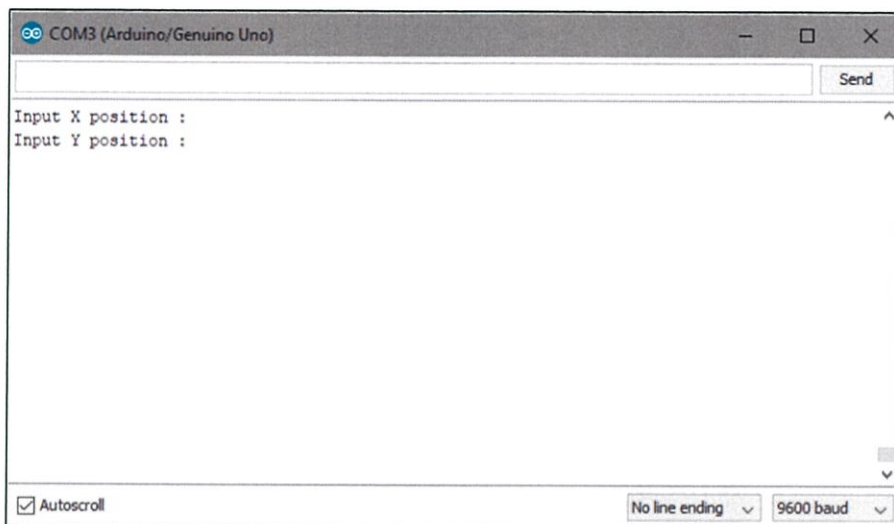
การทดลองออกแบบพร้อมประกอบโครงสร้างเครื่องเปลี่ยนชุดข้อสอบ นำเครื่องมือการออกแบบพร้อมประกอบโครงสร้างเครื่องเปลี่ยนชุดข้อสอบ มาประกอบโครงสร้างเครื่องเปลี่ยนชุดข้อสอบตามที่ได้ออกแบบไว้ โดยเปรียบเทียบเครื่องเปลี่ยนชุดข้อสอบที่ประกอบเสร็จแล้วเทียบกับภาพที่ออกแบบไว้ และสังเกตการใช้งาน

3.3.2 ส่วนชุดเลื่อนตัวกล้องสองแกนและซอฟต์แวร์ควบคุม

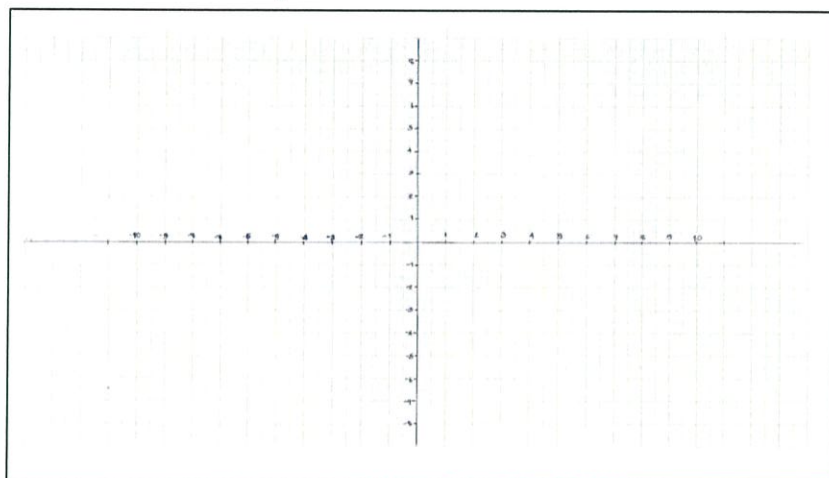
การทดลองความแม่นยำของการเคลื่อนที่ที่ตำแหน่งปลายทางของชุดเลื่อนตัวกล้องสองแกน เริ่มจากติดปากกากับแกนเคลื่อนที่เพื่อเป็นเครื่องมือชี้ตำแหน่งแสดงดังรูปที่ 3.58 แล้วป้อนตำแหน่งที่ต้องการทั้งหมด 4 ชุด ผ่าน Serial Monitor แสดงดังรูปที่ 3.59 โดยชุดแรกให้ตำแหน่งอยู่ในควอดรันต์ที่ 1 แนวแกน X อยู่ระหว่างค่า 0 ถึง 10 กับแนวแกน Y อยู่ระหว่างค่า 0 ถึง 8 ป้อนที่ละตำแหน่ง โดยเริ่มต้นที่ตำแหน่ง (0,0) ไปจนตำแหน่ง (10,8) ชุดที่สองให้ตำแหน่งอยู่ในควอดรันต์ที่ 2 แนวแกน X อยู่ระหว่างค่า 0 ถึง -10 กับแนวแกน Y อยู่ระหว่างค่า 0 ถึง 8 ป้อนที่ละตำแหน่ง โดยเริ่มต้นที่ตำแหน่ง (0,0) ไปจนตำแหน่ง (-10,8) ชุดที่สามให้ตำแหน่งอยู่ในควอดรันต์ที่ 3 แนวแกน X อยู่ระหว่างค่า 0 ถึง -10 กับแนวแกน Y อยู่ระหว่างค่า 0 ถึง -8 ที่ละตำแหน่ง โดยเริ่มต้นที่ตำแหน่ง (0,0) ไปจนตำแหน่ง (-10,-8) และชุดที่สี่ให้ตำแหน่งอยู่ในควอดรันต์ที่ 4 แนวแกน X อยู่ระหว่างค่า 0 ถึง 10 กับแนวแกน Y อยู่ระหว่างค่า 0 ถึง -8 ป้อนที่ละตำแหน่ง โดยเริ่มต้นที่ตำแหน่ง (0,0) ไปจนตำแหน่ง (10,-8) แล้วสังเกตตำแหน่งที่ปลายปากกาซึ่งอยู่ เทียบกับกระดาษกราฟที่แสดงพิกัดจริง โดยลักษณะของกระดาษกราฟแสดงดังรูปที่ 3.60 และบันทึกกระยะรัศมีที่คลาดเคลื่อนของตำแหน่งที่ป้อน



รูปที่ 3.58 การทดลองความแม่นยำของการเคลื่อนที่ที่ตำแหน่งปลายทาง



รูปที่ 3.59 Serial Monitor



รูปที่ 3.60 ลักษณะของกระดาษกราฟ

3.3.3 ชุดเปลี่ยนชุดข้อสอบ และซอฟต์แวร์ควบคุม

การทดลองใช้งานชุดเปลี่ยนชุดข้อสอบ และซอฟต์แวร์ควบคุม เป็นการทดลองใช้งานฟังก์ชันเปลี่ยนชุดข้อสอบที่ประกอบด้วยการทำงานของชุดยกลาดรองชุดข้อสอบ ชุดเปลี่ยนชุดข้อสอบที่ทำงานร่วมกัน ซึ่งถูกควบคุมด้วยซอฟต์แวร์ควบคุม เพื่อทดสอบการใช้งาน และประสิทธิภาพการทำงานของชุดเปลี่ยนชุดข้อสอบ โดยทดลองใช้งานฟังก์ชันเปลี่ยนชุดข้อสอบจำนวน 300 ครั้ง โดยแบ่งการทำงานเป็น 15 รอบการทดลอง และในแต่ละรอบการทดลองจะใส่ชุดข้อสอบจำนวน 20 ชุด สังเกตการทำงานของฟังก์ชันเปลี่ยนชุดข้อสอบ และบันทึกผล

3.3.4 ส่วนโปรแกรมประมวลผลภาพ

3.3.4.1 การทดลองปรับค่าพารามิเตอร์ของตัวกรอง Bilateral

จากการออกแบบตัวกรอง Bilateral มี 3 พารามิเตอร์ที่ต้องกำหนดคือ ขนาดตัวกรอง sigma color และ sigma space ดังนั้นเพื่อหาค่าพารามิเตอร์ที่สามารถลดสัญญาณรบกวนได้มากที่สุด และยังคงรักษาความคมชัดของขอบตัวเลขไว้ได้ จึงออกแบบการทดลองตามขั้นตอนดังนี้

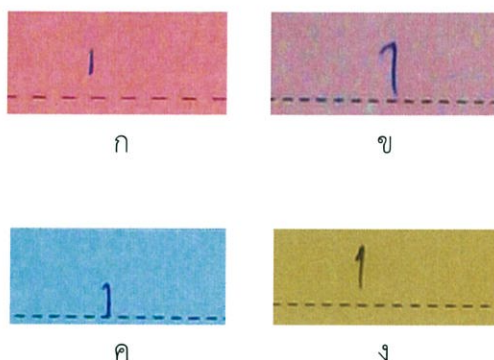
1. นำรูปที่ 3.61 เข้าสู่ตัวกรอง Bilateral
2. ปรับขนาดของตัวกรอง โดยให้ค่า sigma color และ sigma space เป็นค่าคงที่
3. เปรียบเทียบผลความแตกต่างของแต่ละขนาดตัวกรอง
4. ปรับ sigma color โดยนำขนาดของตัวกรองที่ได้ผลลัพธ์ดีที่สุดมาใช้ และกำหนดให้ sigma space ใช้ค่าคงที่
5. เปรียบเทียบผลที่ได้จากการปรับ sigma color
6. ปรับ sigma space โดยนำขนาดของตัวกรองและ sigma color ที่ได้ผลลัพธ์ดีที่สุดมาใช้



รูปที่ 3.61 ภาพสำหรับการทดลองปรับค่าพารามิเตอร์ของตัวกรอง Bilateral

3.3.4.2 การทดลองหาค่าเทรชโฮลด์

จากการออกแบบในกระบวนการเทรชโฮลด์ต้องกำหนดค่าเทรชโฮลด์สำหรับตัดสีนแบ่งส่วนของพื้นหลังกับตัวเลขลายมือ ดังนั้นเพื่อกำหนดค่าเทรชโฮลด์ที่เหมาะสมจึงทดลองใช้สีของกระดาษพื้นหลังที่แตกต่างกัน และทดลองนำภาพจากพื้นหลังที่แตกต่างกันไปพล็อตกราฟฮิสโตแกรมเพื่อพิจารณาเลือกค่าเทรชโฮลด์ที่สามารถแยกตัวเลขลายมือกับพื้นหลังได้ โดยใช้สีพื้นกระดาษทั้งหมด 4 สี คือ สีแดง สีชมพู สีฟ้า และสีเหลือง ซึ่งเป็นสีของชุดข้อสอบทั้งหมดของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล) โดยภาพที่ใช้ในการทดลองเป็นภาพที่ผ่านกระบวนการตัดกรอบช่องกรอกคะแนนแล้ว แสดงได้ดังรูปที่ 3.62



รูปที่ 3.62 ภาพสำหรับการทดลองหาค่าเทรซโฮลด์

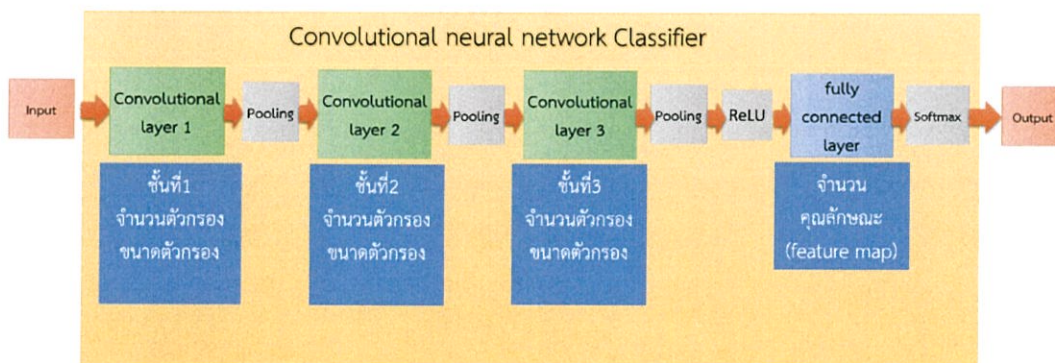
3.3.4.3 การทดสอบการตัดกรอบตัวเลข

การทดสอบการตัดกรอบตัวเลขเป็นการพิจารณาผลการ Find contours ซึ่งจะได้ผลเป็นการตัดกรอบเฉพาะตัวเลข โดยพิจารณาความถูกต้องของการตัดกรอบตัวเลขเป็นร้อยละจากตัวอย่างลายมือตัวเลขจำนวน 1,000 ตัวอย่าง แบ่งเป็นตัวเลขละ 100 ตัวอย่าง

3.3.5 ส่วนโปรแกรมจำแนกตัวเลขด้วยกระบวนการเรียนรู้และจดจำแบบ CNN

3.3.5.1 การทดลองปรับตัวกรองถ่วงน้ำหนัก

การทดลองปรับตัวกรองถ่วงน้ำหนักสามารถปรับได้ 4 ชั้น โดยแบ่งเป็นชั้น convolution ตั้งแต่ชั้นที่ 1 ถึง 3 และชั้น fully connected ตามที่ออกแบบไว้ซึ่งสามารถอธิบายตัวแปรที่สามารถปรับได้ดังรูปที่ 3.63



รูปที่ 3.63 ภาพประกอบการปรับตัวกรองถ่วงน้ำหนัก

จากรูปที่ 3.63 จะพบว่าในแต่ละชั้นของ convolution สามารถปรับค่าได้ 2 ส่วนคือ จำนวนตัวกรองถ่วงน้ำหนัก และขนาดของตัวกรองถ่วงน้ำหนัก และในชั้น fully connected สามารถปรับจำนวนคุณลักษณะได้ ดังนั้นจึงออกแบบการทดลองได้ดังนี้

1. ทดลองปรับขนาดของตัวกรองถ่วงน้ำหนักทุกชั้นตั้งแต่ขนาด 8×8 ถึง 3×3 โดยกำหนดจำนวนตัวกรองถ่วงน้ำหนักเท่ากับ 8 ในชั้น convolution ทั้งหมด และจำนวนคุณลักษณะในชั้น fully connected เท่ากับ 512

2. นำค่าของขนาดของตัวกรองถ่วงน้ำหนักที่มีค่าความถูกต้องสูงที่สุด มาทดลองปรับจำนวนตัวกรองถ่วงน้ำหนักในแต่ละชั้น convolution

โดยชุดข้อมูลตัวเลขที่ใช้สำหรับการเรียนรู้และทดสอบผลการเรียนรู้ เป็นการจัดเก็บตัวอย่างลายมือตัวเลขจากนักศึกษาของ สจล จำนวน 10,000 ตัวอย่าง ซึ่งถูกแบ่งเป็นชุดสำหรับเรียนรู้ 8,000 ตัวอย่าง และชุดทดสอบ 2,000 ตัวอย่าง

3.3.5.2 การทดสอบการจำแนกตัวเลขจากชุดข้อสอบ

การทดสอบการจำแนกตัวเลขจากชุดข้อสอบเป็นทดสอบการใช้งานโปรแกรมจำแนกตัวเลข ด้วยกระบวนการเรียนรู้และจดจำแบบ CNN กับชุดข้อสอบตัวอย่าง จำนวน 100 ชุด ซึ่งแต่ละชุดมีตัวเลข 0 ถึง 9

3.3.6 ส่วนโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้

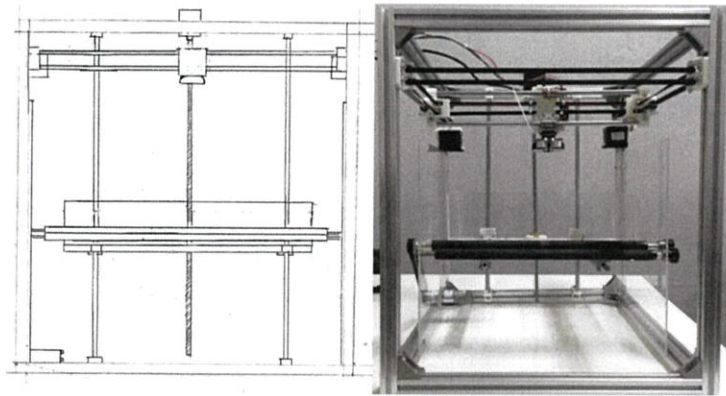
การทำงานของโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ สามารถตรวจสอบได้ว่าผลที่ได้จากการทำงานตรงกับการออกแบบหน้าต่างหรือไม่ ในหน้าต่างเลื่อนตัวกล่องและหน้าต่างตรวจจับตัวเลข โปรแกรมสามารถส่งค่าคงที่ไปยังโปรแกรมเลื่อนตัวกล่องและโปรแกรมควบคุมมอเตอร์ป้อนชุดข้อสอบออก ในหน้าต่างลากกรอบตรวจจับตัวเลข โปรแกรมสามารถลากกรอบบริเวณที่ผู้ใช้ต้องการตรวจจับตัวเลขลายมือ และโปรแกรมสามารถส่งค่าคงที่ไปยังโปรแกรมควบคุมถาดรองชุดข้อสอบได้หรือไม่

บทที่ 4

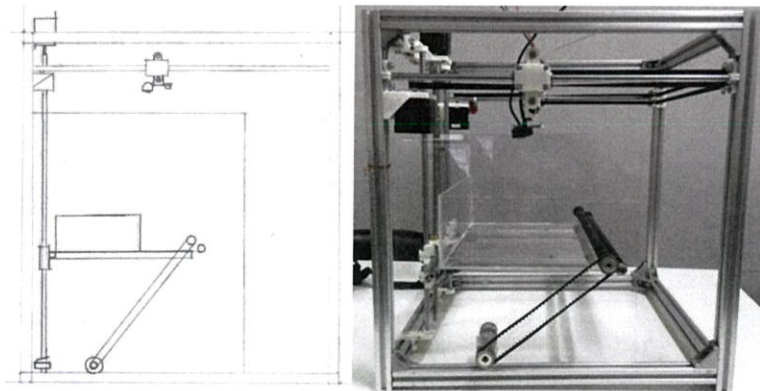
ผลการทดลอง

4.1 การประกอบเครื่องเปลี่ยนชุดข้อสอบ

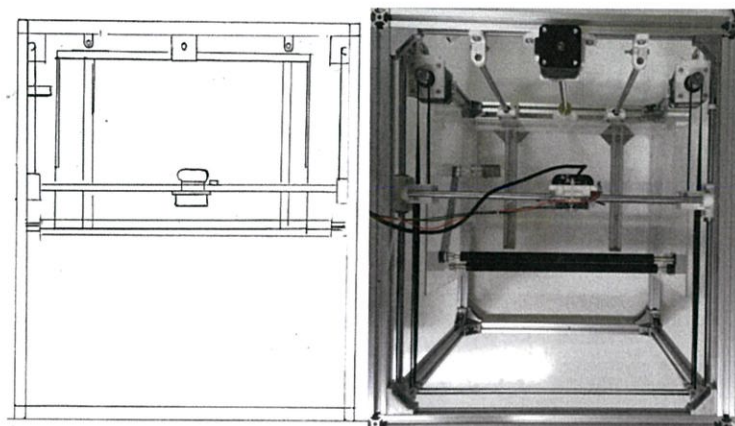
การทดลองออกแบบพร้อมประกอบโครงสร้างเครื่องเปลี่ยนชุดข้อสอบ สามารถประกอบโครงสร้างชุดเครื่องเปลี่ยนชุดข้อสอบได้ตามที่ออกแบบไว้ และสามารถใช้งานได้จริง โดยแสดงโครงสร้างเครื่องเปลี่ยนชุดข้อสอบ ที่ประกอบเทียบกับที่ออกแบบไว้ดังรูปที่ 4.1 4.2 และ 4.3



รูปที่ 4.1 โครงสร้างเครื่องเปลี่ยนชุดข้อสอบเทียบกับที่ออกแบบไว้ มุมมองด้านหน้า



รูปที่ 4.2 โครงสร้างเครื่องเปลี่ยนชุดข้อสอบเทียบกับที่ออกแบบไว้ มุมมองด้านข้าง



รูปที่ 4.3 โครงสร้างเครื่องเปลี่ยนชุดข้อสอบเทียบกับที่ออกแบบไว้ มุมมองด้านบน

จากผลการออกแบบพร้อมประกอบโครงสร้างเครื่องเปลี่ยนชุดข้อสอบ แสดงให้เห็นว่า โครงสร้างเครื่องเปลี่ยนชุดข้อสอบ ที่ออกแบบไว้สามารถใช้งานได้จริง มีความแข็งแรง และสามารถรองรับอุปกรณ์ในส่วนของชุดเลื่อนตัวกล้องสองแกน อุปกรณ์ในส่วนของชุดยกถาดรองชุดข้อสอบ และอุปกรณ์ในส่วนของชุดเปลี่ยนชุดข้อสอบ

4.2 ส่วนชุดเลื่อนตัวกล้องสองแกนและซอฟต์แวร์ควบคุม

ผลการทำงานของชุดเลื่อนตัวกล้องสองแกน จากการป้อนตำแหน่ง 2 ชุด ผ่าน Serial Monitor โดยชุดแรกให้ตำแหน่งอยู่ในควอดรันต์ที่ 1 แนวแกน X อยู่ระหว่างค่า 0 ถึง 10 กับแนวแกน Y อยู่ระหว่างค่า 0 ถึง 8 ป้อนทีละตำแหน่ง โดยเริ่มต้นที่ตำแหน่ง (0,0) ไปจนตำแหน่ง (10,8) ชุดที่สองให้ตำแหน่งอยู่ในควอดรันต์ที่ 2 แนวแกน X อยู่ระหว่างค่า 0 ถึง -10 กับแนวแกน Y อยู่ระหว่างค่า 0 ถึง 8 ป้อนทีละตำแหน่ง โดยเริ่มต้นที่ตำแหน่ง (0,0) ไปจนตำแหน่ง (-10,8) ชุดที่สามให้ตำแหน่งอยู่ในควอดรันต์ที่ 3 แนวแกน X อยู่ระหว่างค่า 0 ถึง -10 กับแนวแกน Y อยู่ระหว่างค่า 0 ถึง -8 ทีละตำแหน่ง โดยเริ่มต้นที่ตำแหน่ง (0,0) ไปจนตำแหน่ง (-10,-8) และชุดที่สี่ให้ตำแหน่งอยู่ในควอดรันต์ที่ 4 แนวแกน X อยู่ระหว่างค่า 0 ถึง 10 กับแนวแกน Y อยู่ระหว่างค่า 0 ถึง -8 ป้อนทีละตำแหน่ง โดยเริ่มต้นที่ตำแหน่ง (0,0) ไปจนตำแหน่ง (10,-8) ทั้งหมด 356 ตัวอย่าง ซึ่งการทดลองในควอดรันต์ที่ 1 2 3 และ 4 ผลการทดลองแสดงดังตารางที่ 4.1 4.2 4.3 และ 4.4 แล้วสังเกตตำแหน่งที่ปลายปากกาซึ่งอยู่ เทียบกับกระดาษกราฟที่แสดงพิกัดจริง และบันทึกกระยะรัศมีที่คลาดเคลื่อนของตำแหน่งที่ป้อน

ตารางที่ 4.1 บันทึกกระยะรัศมีที่คลาดเคลื่อนของตำแหน่งที่ป้อนในควอดดรันต์ที่ 1

พิกัดในแนวแกน Y (ซม.)	8	0	0.05	0	0	0	0	0	0	0	0.05	0.05
	7	0	0	0	0	0	0	0	0	0	0.05	0.05
	6	0	0	0	0	0	0	0	0	0	0.05	0.05
	5	0	0	0	0	0	0	0.05	0.05	0	0	0
	4	0.05	0	0	0	0.05	0	0	0	0.05	0	0
	3	0	0	0	0	0.05	0	0	0	0.05	0	0
	2	0.05	0	0.05	0.05	0.05	0.05	0.05	0.05	0	0.05	0
	1	0	0.05	0.1	0.05	0.05	0.05	0.05	0.05	0	0.05	0
	0	0	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0	0	0
		0	1	2	3	4	5	6	7	8	9	10
พิกัดในแนวแกน X (ซม.)												

ตารางที่ 4.2 บันทึกกระยะรัศมีที่คลาดเคลื่อนของตำแหน่งที่ป้อนในควอดดรันต์ที่ 2

0	0	0	0	0	0	0.05	0	0	0	0	0	8
0	0	0	0	0	0	0.05	0	0.05	0.05	0	0	7
0	0	0	0	0	0	0.05	0.05	0	0	0	0	6
0.05	0.05	0.05	0	0	0	0.1	0	0	0	0	0	5
0	0.05	0	0	0.05	0	0.05	0	0	0	0.05	0	4
0	0.05	0	0	0.05	0	0	0	0	0	0	0	3
0	0	0.05	0.05	0.05	0	0	0	0.05	0.05	0.05	0	2
0	0	0	0	0	0.05	0.1	0.05	0.05	0.05	0	0	1
0	0	0.1	0.05	0	0	0.05	0.05	0.05	0.05	0	0	0
-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0		
พิกัดในแนวแกน X (ซม.)												พิกัดในแนวแกน Y (ซม.)

ตารางที่ 4.3 บันทึกกระยะรัศมีที่คลาดเคลื่อนของตำแหน่งที่ป้อนในควอดดรนต์ที่ 3

พิกัดในแนวแกน X (ซม.)

-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	
0	0	0.1	0.05	0	0	0.05	0.05	0.05	0.05	0	0
0.05	0.05	0.1	0.05	0.05	0.05	0	0	0	0	0.05	-1
0	0	0	0.05	0.1	0.05	0	0.05	0	0	0.05	-2
0	0.05	0.05	0.05	0.05	0.05	0.05	0	0	0	0	-3
0.05	0.05	0.05	0.05	0.1	0.1	0	0	0	0	0.05	-4
0.05	0.05	0.1	0.05	0.1	0.05	0	0	0	0	0.05	-5
0	0	0.1	0	0.1	0	0	0	0	0	0	-6
0	0.1	0.1	0.1	0.05	0	0	0.05	0	0	0	-7
0	0.05	0.1	0.05	0	0	0	0	0	0	0.05	-8

พิกัดในแนวแกน Y (ซม.)

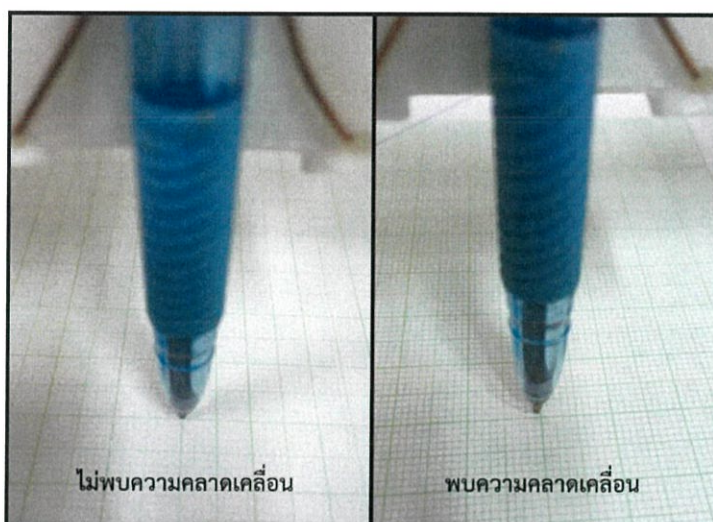
ตารางที่ 4.4 บันทึกกระยะรัศมีที่คลาดเคลื่อนของตำแหน่งที่ป้อนในควอดดรนต์ที่ 4

พิกัดในแนวแกน X (ซม.)

	0	1	2	3	4	5	6	7	8	9	10
0	0	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0	0	0
-1	0.05	0	0	0.05	0.05	0.05	0	0	0	0	0.05
-2	0.05	0	0	0	0.05	0	0	0.05	0.1	0	0.05
-3	0	0	0	0	0.05	0	0	0.05	0.05	0	0.1
-4	0.05	0	0	0	0	0	0	0.05	0.1	0	0
-5	0.05	0.05	0.05	0.05	0	0	0	0.05	0.05	0	0
-6	0	0	0.05	0	0.05	0.05	0	0	0	0	0
-7	0	0	0.05	0	0	0	0	0	0	0	0
-8	0.05	0	0	0	0.05	0.05	0.05	0	0	0	0

พิกัดในแนวแกน Y (ซม.)

จากการทดลองจากตารางที่ 4.1 4.2 4.3 และ 4.4 พบตัวอย่างที่คลาดเคลื่อนทั้งหมด 142 ตัวอย่าง จากทั้งหมด 356 ตัวอย่าง คิดเป็นร้อยละ 39.89 และพบความคลาดเคลื่อนอยู่ในช่วง 0.05 เซนติเมตรถึง 0.1 เซนติเมตรแสดงดังรูปที่ 4.4 คิดเป็นความคลาดเคลื่อนเฉลี่ยได้ 0.0579 เซนติเมตร ปัจจัยที่ทำให้เกิดความคลาดเคลื่อนคือ อุปกรณ์ Stepping Motor ที่เลือกมาใช้งานมีคุณสมบัติ Step angle accuracy เท่ากับ $\pm 5\%$ จึงทำให้มีโอกาสจะเกิดความคลาดเคลื่อนขึ้นได้ และจากคุณสมบัติ Step angle เท่ากับ 1.8 องศา ต้องหมุนไป 90 องศาหรือ 50 ก้าว จึงจะทำให้ Stepping Motor หมุนได้เป็นระยะ 1 เซนติเมตร จะมีความความคลาดเคลื่อนเป็น 4.5 องศา หรือเทียบได้ 0.05 เซนติเมตร จากการนำ Step angle accuracy เท่ากับ $\pm 5\%$ ไปคูณ



รูปที่ 4.4 การทดลองที่พบความคลาดเคลื่อน

4.3 ส่วนชุดเปลี่ยนชุดข้อสอบและซอฟต์แวร์ควบคุม

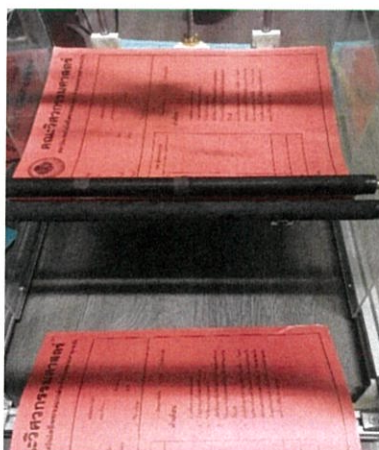
การทดลองใช้งานชุดเปลี่ยนชุดข้อสอบ และซอฟต์แวร์ควบคุม เป็นการทดลองใช้งาน ฟังก์ชันเปลี่ยนชุดข้อสอบที่ประกอบด้วยการทำงานของชุดยกถาดรองชุดข้อสอบ ชุดเปลี่ยนชุดข้อสอบที่ทำงานร่วมกัน ซึ่งถูกควบคุมด้วยซอฟต์แวร์ควบคุม เพื่อทดสอบการใช้งาน และประสิทธิภาพการทำงานของชุดเปลี่ยนชุดข้อสอบ โดยทดลองใช้งานฟังก์ชันเปลี่ยนชุดข้อสอบ จำนวน 400 ครั้ง โดยแบ่งการทำงานเป็น 20 รอบการทดลอง และในแต่ละรอบการทดลองจะใส่ชุดข้อสอบจำนวน 20 ชุด สังเกตการทำงานของฟังก์ชันเปลี่ยนชุดข้อสอบ โดยผลการทดลองแสดงดังตารางที่ 4.5

ตารางที่ 4.5 ปัญหาที่พบจากการใช้งานชุดเปลี่ยนชุดข้อสอบ

ปัญหาที่พบ	จำนวนครั้งที่เกิดปัญหา
ชุดข้อสอบไม่เลื่อนออกมา	30
ชุดข้อสอบเลื่อนออกมาได้ไม่หมดทั้งชุด	21
ชุดข้อสอบเลื่อนติดกันออกมา 2 ชุด	27

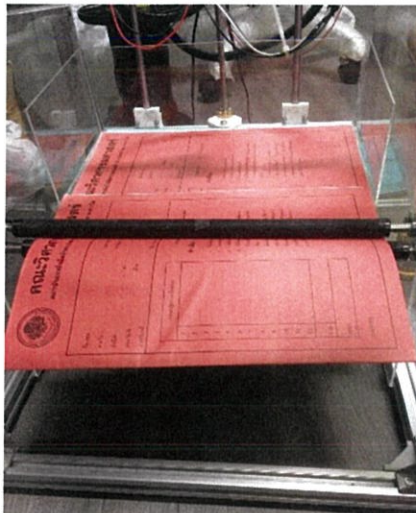
จากตารางที่ 4.5 พบจำนวนครั้งที่เกิดปัญหาทั้งหมด 78 ครั้ง จากการทดลองทั้งหมด 400 ครั้ง โดยพบปัญหาทั้งหมด 3 กรณีคือ ชุดข้อสอบไม่เลื่อนออกมาเกิดขึ้น 30 ครั้ง ชุดข้อสอบเลื่อนออกมาได้ไม่หมดทั้งชุดเกิดขึ้น 21 ครั้ง และชุดข้อสอบเลื่อนติดกันออกมา 2 ชุดเกิดขึ้น 27 ครั้ง ดังนั้นฟังก์ชันเปลี่ยนชุดข้อสอบสามารถเปลี่ยนชุดข้อสอบสำเร็จคิดเป็นร้อยละ 80.5

กรณีชุดข้อสอบไม่เลื่อนออกมาแสดงดังรูปที่ 4.5 มีปัจจัยสำคัญคือ อุปกรณ์ลูกยางซึ่งทำหน้าที่ดึงชุดข้อสอบออกจากถาดรองชุดข้อสอบ เพราะความเหนียวของลูกยางต่ำ จึงไม่สามารถดึงชุดข้อสอบออกมาได้ และถาดรองชุดข้อสอบอยู่ในระยะที่ต่ำกว่าระยะเปลี่ยนชุดข้อสอบ ทำให้ลูกยางไม่สัมผัสกับชุดข้อสอบ



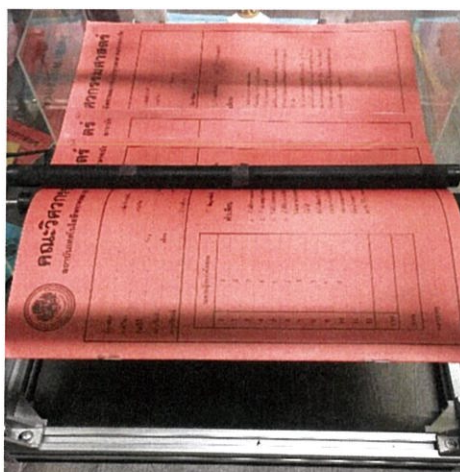
รูปที่ 4.5 ชุดข้อสอบไม่เลื่อนออกมา

กรณีชุดข้อสอบเลื่อนออกมาได้ไม่หมดทั้งชุดแสดงดังรูปที่ 4.6 ปัจจัยสำคัญคือ ลูกยาง สัมผัสกับชุดข้อสอบเพียงเล็กน้อย ทำให้ชุดข้อสอบถูกดึงออกมาเพียงเล็กน้อย



รูปที่ 4.6 ชุดข้อสอบเลื่อนออกมาได้ไม่หมดทั้งชุด

และกรณีชุดข้อสอบเลื่อนติดกันออกมา 2 ชุดแสดงดังรูปที่ 4.7 ปัจจัยสำคัญคือ ระยะของถาดรองชุดข้อสอบสูงกว่าระยะเลื่อนชุดข้อสอบ ทำให้ชุดข้อสอบในถาดรองชุดถูกอัดแน่นกับลูกยาง ชุดข้อสอบจึงติดกันออกมา






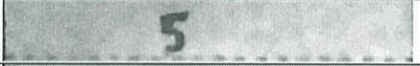


รูปที่ 4.7 ชุดข้อสอบเลื่อนติดกันออกมา 2 ชุด

4.4 ส่วนโปรแกรมประมวลผลภาพ

4.4.1 การทดลองปรับค่าพารามิเตอร์ของตัวกรอง Bilateral




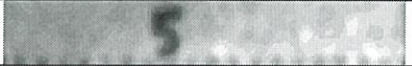
จากการทดลองปรับค่าพารามิเตอร์ของตัวกรอง Bilateral โดยใช้อินพุตเป็นรูปที่ 3.61 โดยทดลองกำหนดค่า sigma color เท่ากับ 5 sigma space เท่ากับ 5 และทดลองปรับขนาดตัวกรอง ซึ่งผลการทดลองสามารถแสดงได้ในตารางที่ 4.6

ตารางที่ 4.6 ผลการทดลองปรับขนาดของตัวกรอง Bilateral

ขนาดตัวกรอง	ผลจากตัวกรอง Bilateral
5	
15	
25	
35	
45	
55	

จากตารางที่ 4.6 พบว่าเมื่อเพิ่มขนาดตัวกรองมากขึ้น สัญญาณรบกวนในภาพจะลดน้อยลง สังเกตได้จากพื้นหลังภาพที่เรียบขึ้น แต่เมื่อขนาดตัวกรองเท่า 25 เป็นต้นไปจะพบว่าภาพไม่มีการเปลี่ยนแปลง ดังนั้นขนาดตัวกรองที่พิจารณาว่าเหมาะสมเท่ากับ 25 ทดลองปรับค่า sigma color และกำหนดค่าขนาดตัวกรองเท่ากับ 25 และ sigma space เท่ากับ 5 ซึ่งผลการทดลองสามารถแสดงได้ในตารางที่ 4.7




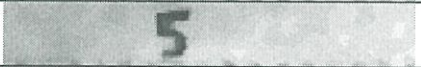

ตารางที่ 4.7 ผลการทดลองปรับค่า sigma color ของตัวกรอง Bilateral

sigma color	ผลจากตัวกรอง Bilateral
5	
15	
25	
35	

จากตารางที่ 4.7 พบว่าเมื่อปรับค่า sigma color เพิ่มขึ้นจะทำให้ภาพทั้งหมดเบลอ และทำให้ขอบของตัวเลขไม่คมชัด และเริ่มกลมกลืนกับพื้นหลัง จึงพิจารณาเลือกค่าต่ำที่สุดที่ตัวกรองสามารถกำหนดได้คือเท่ากับ 5

ทดลองปรับค่า และ sigma space และกำหนดค่าขนาดตัวกรองเท่ากับ 25 sigma color เท่ากับ 5 ซึ่งผลการทดลองสามารถแสดงได้ในตารางที่ 4.8

ตารางที่ 4.8 ผลการทดลองปรับค่า sigma space ของตัวกรอง Bilateral

sigma space	ผลจากตัวกรอง Bilateral
5	
15	
25	
35	
45	

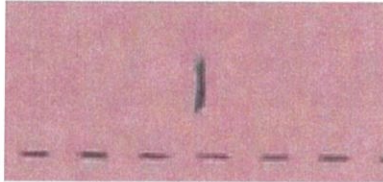
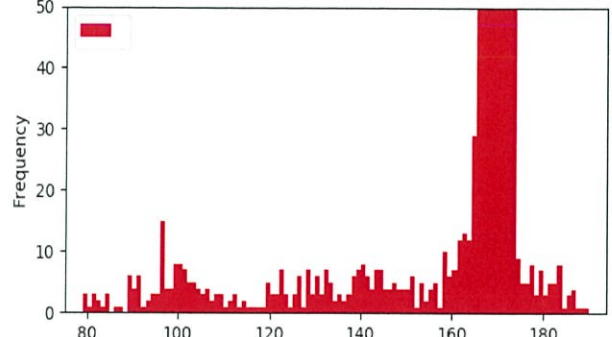
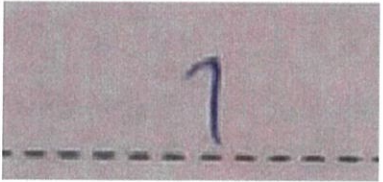
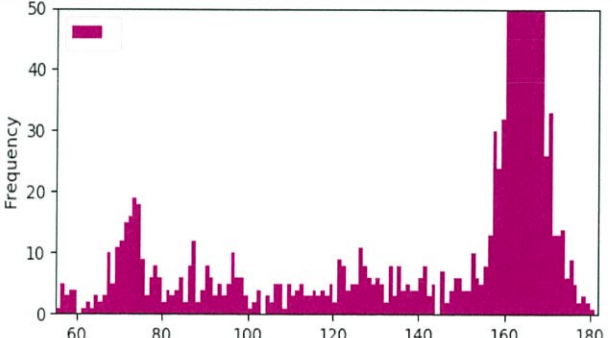
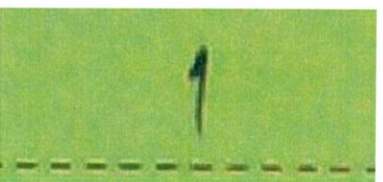
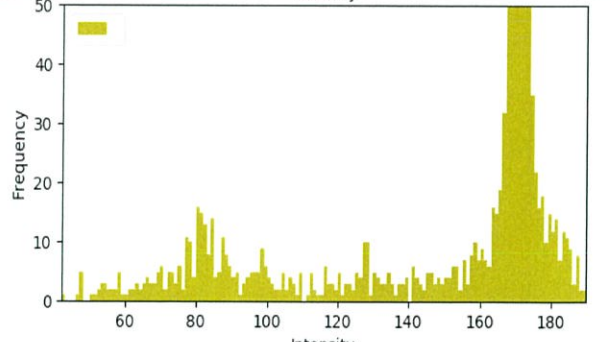
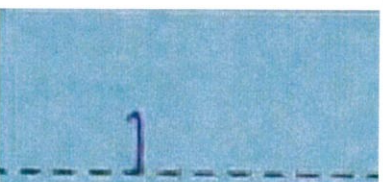
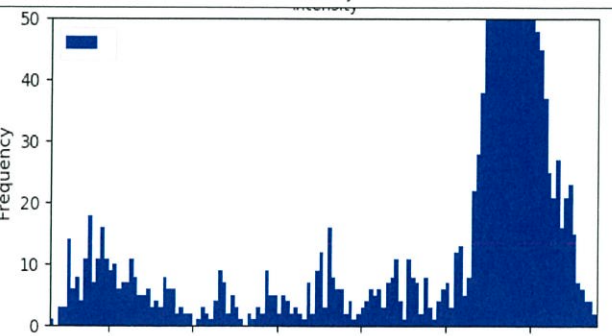
จากตารางที่ 4.8 พบว่าเมื่อปรับค่า sigma space ภาพมีความสว่างเพิ่มขึ้นเล็กน้อย และไม่มีผลต่อการลดสัญญาณรบกวน จึงพิจารณาเลือกค่า sigma space ต่ำที่สุดที่ตัวกรองสามารถกำหนดได้คือเท่ากับ 5

ดังนั้นจากการทดลองสามารถกำหนดค่าพารามิเตอร์ของตัวกรอง Bilateral ที่เหมาะสมกับการตัดกรอบตัวเลขได้ โดยกำหนดค่าขนาดของตัวกรองเท่ากับ 25 sigma color เท่ากับ 5 และ sigma space เท่ากับ 5

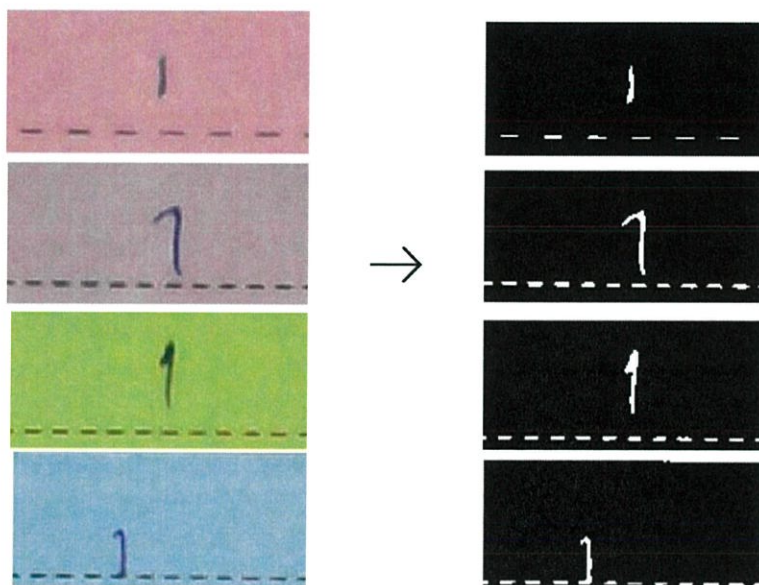
4.4.2 การทดลองหาค่าเทรชโฮลด์

จากการทดลองหาค่าเทรชโฮลด์ โดยใช้การพล็อตกราฟฮิสโตแกรมในการเลือกค่าเทรชโฮลด์ โดยใช้ภาพอินพุตเป็นชุดข้อสอบของ สจล ทั้ง 4 สี ซึ่งการทดลองจะนำภาพไปแปลงเป็นภาพขาวดำและพล็อตกราฟฮิสโตแกรมของความเข้มพิกเซล (intensity) ซึ่งผลการทดลองสามารถแสดงได้ในตารางที่ 4.9

ตารางที่ 4.9 ผลการพล็อตกราฟฮิสโตแกรมของพื้นหลังแต่ละสี

ภาพต้นฉบับ	ผลการพล็อตกราฟฮิสโตแกรม
	
	
	
	

จากตารางที่ 4.9 จะพบว่าจากกราฟฮิสโตแกรมทั้ง 4 กราฟ มีการกระจายตัวเป็น 2 กลุ่ม แบ่งเป็นกลุ่มความเข้มที่มีความถี่สูงเป็นพื้นหลัง และกลุ่มความเข้มที่มีความถี่ต่ำเป็นตัวเลข เมื่อพิจารณาจะพบว่าการกระจายตัวของทั้งสองกลุ่มมีค่าใกล้เคียงกันมากในแต่ละกราฟ ดังนั้นสามารถค่าเทรชโฮลด์ที่ใช้ร่วมกันได้ทั้ง 4 สีพื้นหลัง ซึ่งเมื่อพิจารณากราฟจะพบว่าค่ากึ่งกลางของการกระจายตัวของทั้งสองกลุ่มในทุกสีพื้นหลังอยู่ที่ความเข้มเท่ากับ 110 จึงเลือกค่าเทรชโฮลด์เท่ากับ 110 และเมื่อทดลองนำค่าดังกล่าวไปใช้ในกระบวนการเทรชโฮลด์จะได้ผลดังรูปที่ 4.8



รูปที่ 4.8 ผลจากกระบวนการเทรชโฮลด์ที่ค่าเทรชโฮลด์เท่ากับ 110

จากรูปที่ 4.8 จะพบว่าผลจากผ่านกระบวนการเทรชโฮลด์สามารถแบ่งแยกตัวเลขและพื้นหลังได้ชัดเจน

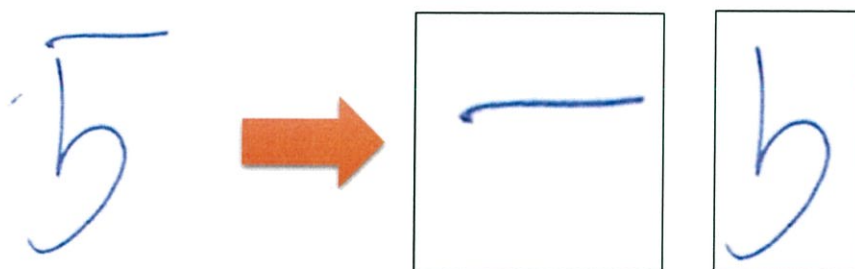
4.4.3 การทดสอบการตัดกรอบตัวเลข

จากการทดสอบการตัดกรอบตัวเลขด้วยวิธีการ Find contours โดยใช้ตัวอย่างลายมือตัวเลขจำนวน 1,000 ตัวอย่าง พบว่าผลการตัดกรอบมีความถูกต้องคิดเป็นร้อยละ 96.1 ซึ่งแต่ละตัวเลขมีค่าความถูกต้องที่แตกต่างแสดงได้ดังตารางที่ 4.10

ตารางที่ 4.10 ผลการทำงานของตัดกรอบและตัวเลขลาย

ตัวเลข	ผลการตัดกรอบตัวเลขลายมือได้ถูกต้อง (ร้อยละ)
0	96
1	93
2	96
3	97
4	96
5	89
6	99
7	98
8	98
9	99

จากตารางที่ 4.10 พบว่าตัวเลขที่ตัดกรอบได้ถูกต้องมากที่สุดคือตัวเลข 6 กับ 9 และตัวเลขที่ตัดกรอบผิดพลาดมากที่สุดคือตัวเลข 5 1 และ 2 ตามลำดับ ดังนั้นจึงนำภาพตัวเลขที่ตัดกรอบผิดพลาดมากที่สุดมาพิจารณา ดังรูปที่ 4.9



รูปที่ 4.9 ตัวอย่างการเขียนตัวเลขที่ไม่เป็นเส้นเดียวกัน

จากรูปที่ 4.9 เป็นตัวเลขที่ส่วนใหญ่มักเขียนตัวเลข 5 ไม่เป็นเส้นเดียวกัน ดังนั้นเมื่อเข้ากระบวนการ find contours จะถูกตัดเป็น 2 ส่วน

4.5 ส่วนโปรแกรมจำแนกตัวเลขด้วยกระบวนการเรียนรู้และจดจำแบบ CNN

4.5.1 การทดลองปรับตัวกรองถ่วงน้ำหนัก

การทดลองปรับตัวกรองถ่วงน้ำหนักเป็นการทดลองปรับค่า 2 ส่วนคือ จำนวนตัวกรองถ่วงน้ำหนัก และขนาดของตัวกรองถ่วงน้ำหนักซึ่งผลการทดลองมีดังนี้

1. ผลปรับขนาดของตัวกรองถ่วงน้ำหนักทั้ง 3 ชั้นตั้งแต่ขนาด 8×8 ถึง 3×3 โดยกำหนด

จำนวนตัวกรองถ่วงน้ำหนักเท่ากับ 8 ในชั้น convolution ทั้งหมด และจำนวนคุณลักษณะในชั้น fully connected เท่ากับ 512 สามารถแสดงได้ดังตารางที่ 4.11 ซึ่งผลการทดลองเป็นการใช้ชุดข้อมูลการเรียนรู้และชุดข้อมูลทดสอบชุดเดียวกัน

ตารางที่ 4.11 ผลการทดลองปรับขนาดของตัวกรองถ่วงน้ำหนัก

ขนาดของตัวกรองชั้นที่ 1	ขนาดของตัวกรองชั้นที่ 2	ขนาดของตัวกรองชั้นที่ 3	ความถูกต้องจากชุดทดสอบ (ร้อยละ)
8x8	8x8	8x8	62.40
8x8	8x8	7x7	77.95
8x8	8x8	6x6	78.61
8x8	8x8	5x5	80.81
8x8	8x8	4x4	70.57
8x8	8x8	3x3	78.47
8x8	7x7	7x7	72.28
8x8	7x7	6x6	66.71
8x8	7x7	5x5	75.90
8x8	7x7	4x4	81.23
8x8	7x7	3x3	82.38
8x8	6x6	6x6	79.47
8x8	6x6	5x5	79.40

ตารางที่ 4.11 ผลการทดลองปรับขนาดของตัวกรองถ่วงน้ำหนัก (ต่อ)

ขนาดของตัวกรองชั้นที่ 1	ขนาดของตัวกรองชั้นที่ 2	ขนาดของตัวกรองชั้นที่ 3	ความถูกต้องจากชุดทดสอบ (ร้อยละ)
8x8	6x6	4x4	74.00
8x8	6x6	3x3	76.04
8x8	5x5	5x5	73.30
8x8	5x5	4x4	81.85
8x8	5x5	3x3	79.00
8x8	4x4	4x4	75.95
8x8	4x4	3x3	74.95
8x8	3x3	3x3	78.57
7x7	7x7	7x7	78.85
7x7	7x7	6x6	70.19
7x7	7x7	5x5	67.19
7x7	7x7	4x4	77.28
7x7	7x7	3x3	74.95
7x7	6x6	6x6	83.14
7x7	6x6	5x5	74.61
7x7	6x6	4x4	76.09
7x7	6x6	3x3	74.09
7x7	5x5	5x5	74.52
7x7	5x5	4x4	73.80
7x7	5x5	3x3	79.95
6x6	6x6	6x6	74.47
6x6	6x6	5x5	76.50
6x6	6x6	4x4	74.90
6x6	6x6	3x3	71.76

ตารางที่ 4.11 ผลการทดลองปรับขนาดของตัวกรองถ่วงน้ำหนัก (ต่อ)

ขนาดของตัวกรองชั้นที่ 1	ขนาดของตัวกรองชั้นที่ 2	ขนาดของตัวกรองชั้นที่ 3	ความถูกต้องจากชุดทดสอบ (ร้อยละ)
6x6	5x5	5x5	73.38
6x6	5x5	4x4	79.33
6x6	5x5	3x3	79.33
5x5	5x5	5x5	79.14
5x5	5x5	4x4	80.00
5x5	5x5	3x3	79.00
5x5	4x4	4x4	82.61
5x5	4x4	3x3	83.04
4x4	4x4	4x4	77.52
4x4	4x4	3x3	74.61
4x4	3x3	3x3	83.09
3x3	3x3	3x3	47.28

จากการทดลองพบว่าขนาดของตัวกรองที่มีขนาด 7x7 6x6 และ 6x6 เรียงตามลำดับชั้นจะให้ผลการทดลองที่มีค่าความถูกต้องมากที่สุดเป็นร้อยละ 83.14 จากชุดทดสอบ ซึ่งจากการทดลองสามารถอธิบายได้ว่าขนาดของตัวกรองมีผลกระทบต่อการใช้งานตัวเลขมาก และที่สังเกตได้จากตารางที่ 4.11 คือการกำหนดตัวกรองที่มีขนาดเท่ากันทั้ง 3 ชั้นจะให้ความถูกต้องที่ค่อนข้างต่ำ

2. ผลจากการนำค่าของขนาดของตัวกรองถ่วงน้ำหนักที่มีค่าความถูกต้องสูงที่สุด มาทดลองปรับจำนวนตัวกรองถ่วงน้ำหนักในแต่ละชั้น convolution สามารถแสดงผลการทดลองได้ดังตารางที่ 4.12

ตารางที่ 4.12 ผลการทดลองปรับจำนวนตัวกรองถ่วงน้ำหนักในแต่ละชั้น convolution

จำนวนตัวกรอง ชั้นที่ 1	จำนวนตัวกรอง ชั้นที่ 2	จำนวนตัวกรอง ชั้นที่ 3	ความถูกต้องจากชุด ทดสอบ (ร้อยละ)
8	8	16	84.09
8	16	32	85.04
16	32	64	88.85
32	64	128	81.89
64	128	256	80.16

จากการตารางที่ 4.12 พบว่าจำนวนตัวกรองที่ 16 32 และ 64 เรียงตามลำดับชั้นให้ค่าความถูกต้องจากชุดทดสอบมากที่สุด โดยแนวโน้มในช่วงแรกเมื่อเพิ่มจำนวนตัวกรองจะให้ความถูกต้องสูงขึ้น แต่เมื่อเพิ่มขึ้นจนถึงระดับหนึ่งจะให้ความถูกต้องลดลง

ดังนั้นผลการปรับกรองถ่วงน้ำหนักในชั้นในชั้น convolution สามารถให้ค่าความถูกต้องสูงสุดอยู่ที่ร้อยละ 85.76 โดยมีการกำหนดค่าตัวกรองที่มีขนาด 7×7 6×6 และ 6×6 เรียงตามลำดับกำหนดจำนวนตัวกรองที่ 16 32 และ 64 เรียงตามลำดับ

4.5.1 การทดลองการจำแนกตัวเลขจากชุดข้อสอบ

การทดลองการจำแนกตัวเลขจากชุดข้อสอบจำนวน 100 ชุด พบว่าสามารถจำแนกตัวเลขได้ถูกต้องเป็นร้อยละ 87.8 ซึ่งสามารถพิจารณาแยกเป็นแต่ละตัวเลข 0 ถึง 9 ได้ดังตารางที่ 4.13

ตารางที่ 4.13 ผลการทำงานของตัดกรอบและตัวเลขลาย

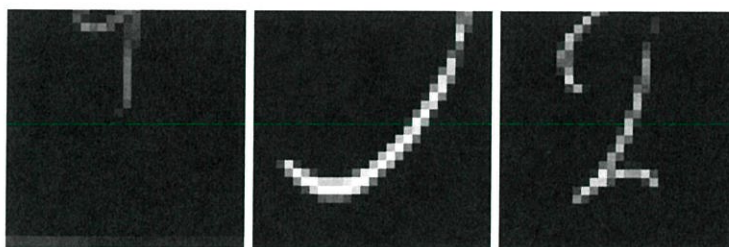
ตัวเลข	จำแนกได้ถูกต้อง (ร้อยละ)
0	94
1	92
2	84

ตารางที่ 4.13 ผลการทำงานของตัดกรอบและตัวเลขลายมือ (ต่อ)

ตัวเลข	จำแนกได้ถูกต้อง (ร้อยละ)
3	92
4	74
5	96
6	84
7	92
8	94
9	76

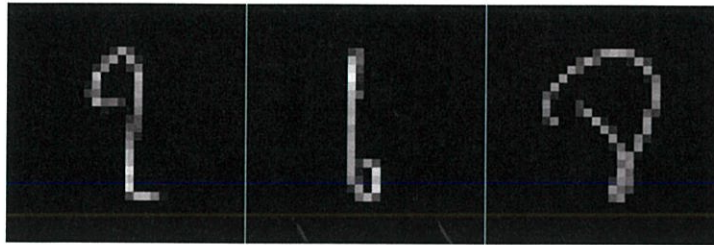
จากตารางที่ 4.13 พบว่าตัวเลขจากลายมือ เลข 9 จำแนกได้ถูกต้องน้อยที่สุด และตัวเลขจากลายมือ เลข 0 และ 8 จำแนกได้ถูกต้องมากที่สุด จากผลการทดลองนำภาพที่จำแนกผิดพลาดมาพิจารณาพบว่าเกิดจาก 2 ปัญหาหลักคือ

1. เกิดจากการตัดกรอบตัวเลขไม่ครบหรือขาดหายไปบางส่วนโดยพบว่าการจำแนกผิดพลาดจากปัญหาการตัดกรอบตัวเลขไม่ครบ ซึ่งสามารถแสดงตัวอย่างได้ดังรูปที่ 4.10



รูปที่ 4.10 การตัดกรอบตัวเลขไม่ครบ

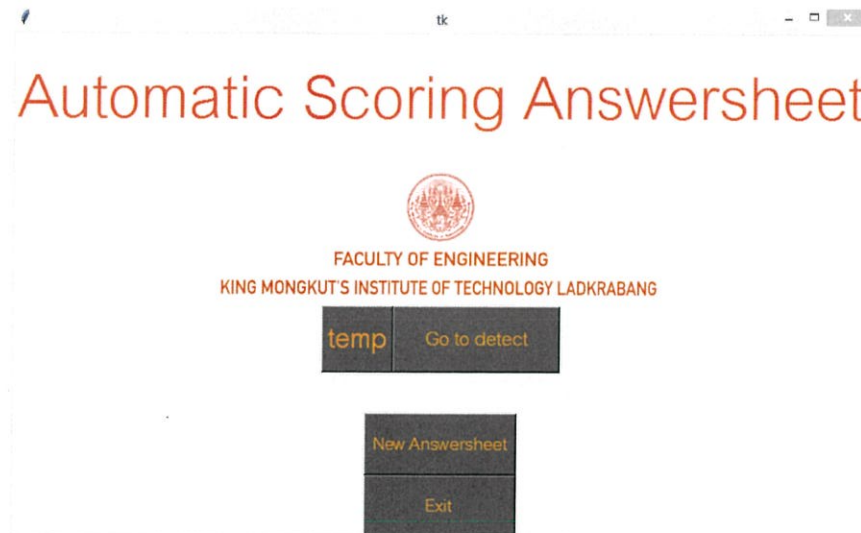
2. เกิดจากลักษณะตัวเลขที่เขียนใกล้เคียงกันแสดงตัวอย่างได้ดังรูปที่ 4.11



รูปที่ 4.11 ลักษณะตัวเลขที่เขียนใกล้เคียง

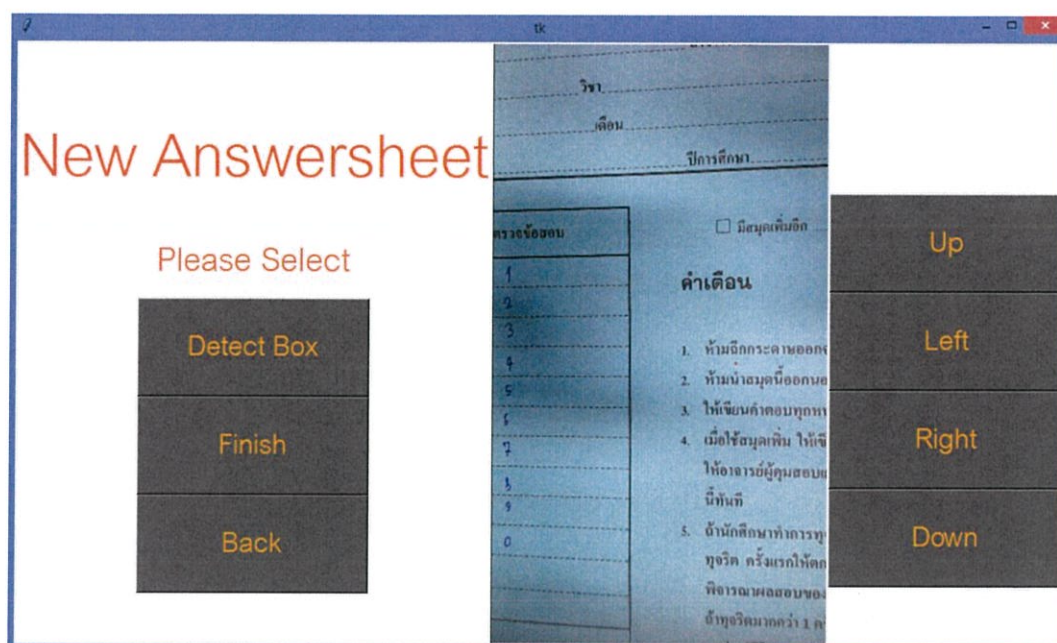
4.6 ส่วนโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้

ผลการทำงานของโปรแกรม GUI พบว่าโปรแกรมเริ่มต้นการทำงาน โดยจะแสดงหน้าต่างเริ่มต้นดังรูปที่ 4.12



รูปที่ 4.12 หน้าต่างเริ่มต้น

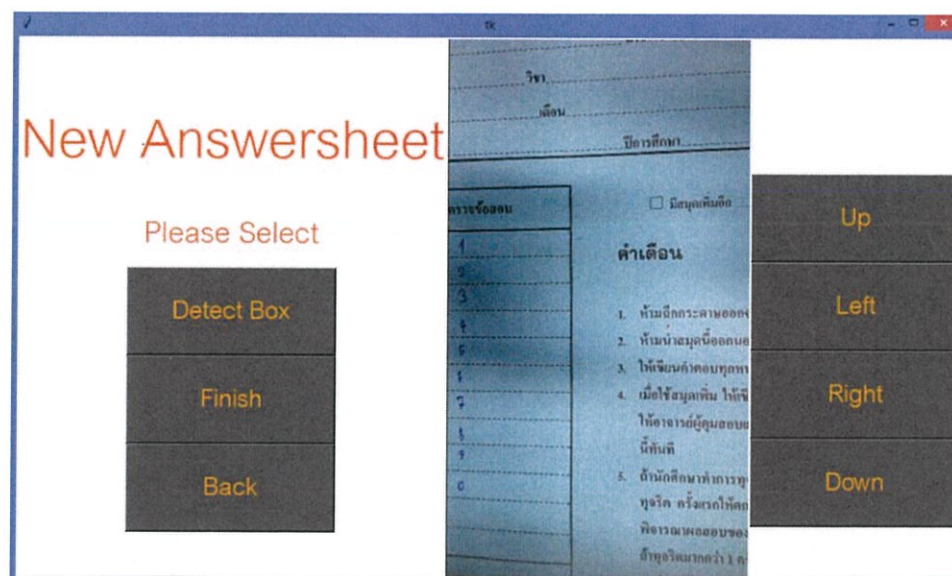
จากรูปที่ 4.12 เมื่อกดปุ่ม “New Answersheet” จะไปยังหน้าต่างเลื่อนตัวกล่อง เพื่อเป็นการตั้งค่าชุดข้อสอบรูปแบบใหม่ที่ผู้ใช้สามารถกำหนดส่วนที่ต้องการลากกรอบตรวจจับ ได้แก่ รหัสนักศึกษา หรือคะแนน เมื่อกดปุ่ม “Back” จะย้อนกลับไปยังหน้าเริ่มต้น



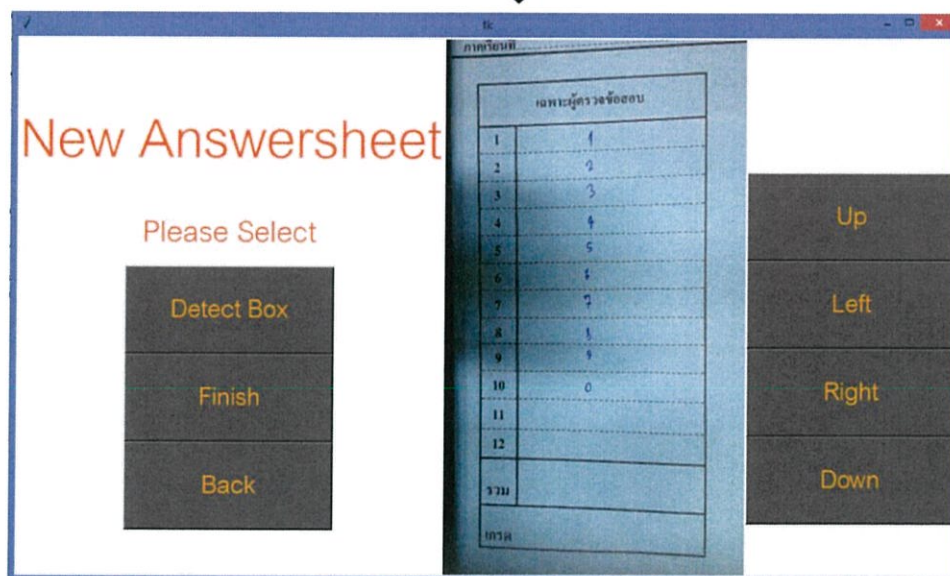
รูปที่ 4.13 ผลที่ได้จากการกดปุ่ม “New Answersheet”

จากรูปที่ 4.13 สังเกตเห็นได้ว่าในหน้าต่างเลื่อนตัวกล้อง ตัวกล้องจะเริ่มต้นอยู่ที่ตำแหน่งกึ่งกลางของตารางชุดข้อสอบ เมื่อกดปุ่ม Detect Box จะไปยังหน้าต่างลากกรอบตรวจจับ เมื่อกดปุ่ม “Finish” จะไปยังหน้าต่างยืนยันชื่อชุดข้อสอบ เมื่อกดปุ่ม “Back” จะย้อนกลับไปไปยังหน้าต่างเริ่มต้น เมื่อกดปุ่ม “Up” “Left” “Right” “Down” โปรแกรมจะสั่งให้ตัวกล้องเลื่อนขึ้น ซ้าย ขวา และลงตามลำดับ ทดลองกดปุ่ม “Down” และ “Right” เพื่อให้สามารถเห็นส่วนที่เป็นคะแนนของชุดข้อสอบสจล.

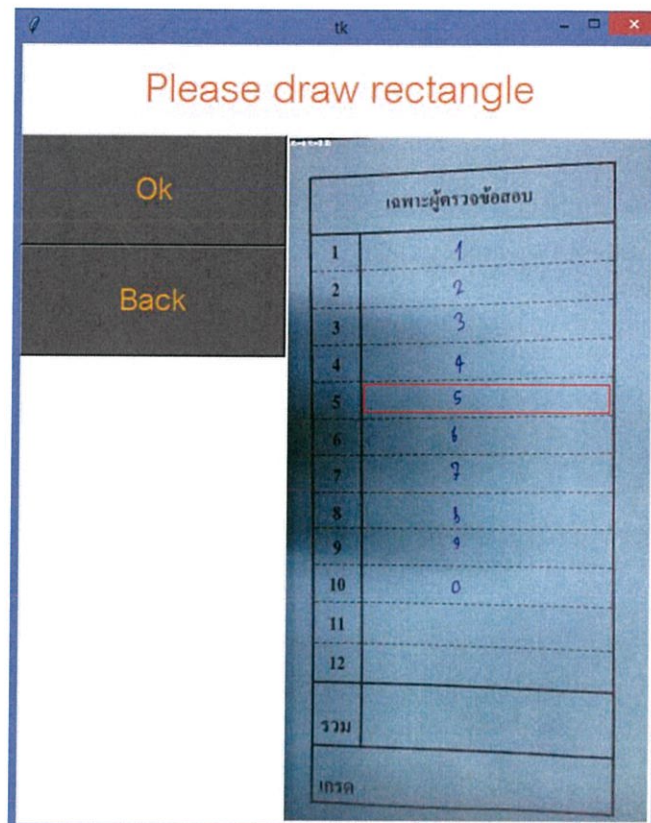
เมื่อกดปุ่ม “Detect Box” โปรแกรมจะหยุดรับภาพจากตัวกล้องและจะไปยังหน้าต่างลากกรอบตรวจจับ ซึ่งสามารถลากกรอบตรวจจับได้ภายในภาพ เมื่อกดปุ่ม “Back” ในหน้าต่างลากกรอบตรวจจับจะย้อนกลับไปไปยังหน้าต่างเลื่อนตัวกล้อง ทดลองลากกรอบตรวจจับบริเวณที่เป็นส่วนของคะแนนแสดงได้ดังรูปที่ 4.14



กดปุ่มเลื่อนตัวกล้อง

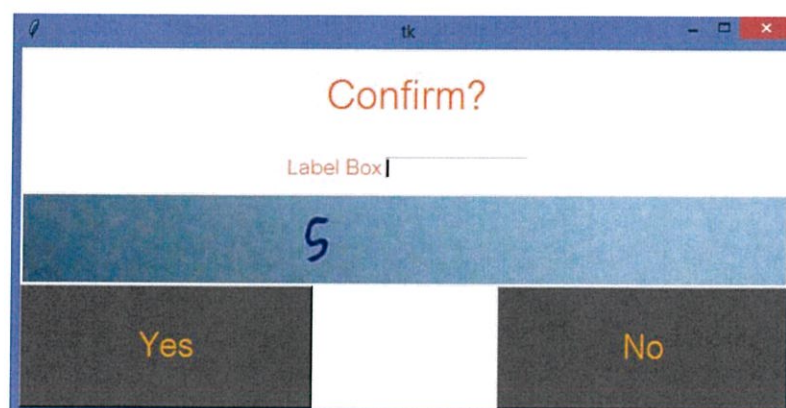


รูปที่ 4.14 ผลที่ได้จากการกดปุ่มเลื่อนตัวกล้อง



รูปที่ 4.15 ผลที่ได้จากการกดปุ่ม “Detect Box” และลากกรอบตรวจจับ

จากรูปที่ 4.15 โปรแกรมสามารถลากกรอบตรวจจับภายในภาพได้ เมื่อกดปุ่ม “Ok” จะไปยังหน้าต่างยืนยันการลากกรอบ เมื่อกดปุ่ม “No” จะย้อนกลับไปหน้าต่างลากกรอบตรวจจับ

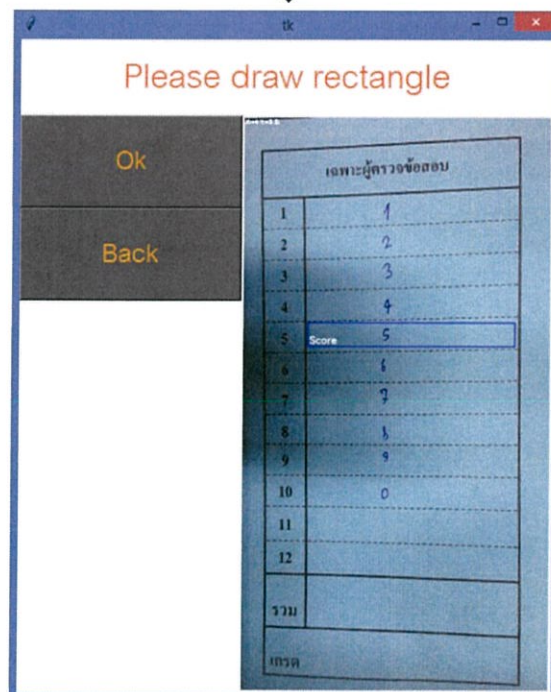


รูปที่ 4.16 ผลที่ได้จากการกดปุ่ม “Ok”

จากรูปที่ 4.16 เมื่อใส่ชื่อส่วนที่ลากกรอบตรวจจับลงในช่องว่างแล้วกดปุ่ม “Yes” จะไปยังหน้าต่างลากกรอบตรวจจับที่มีกรอบที่ผู้ใช้ลากไว้ปรากฏบนภาพเป็นกรอบสีน้ำเงินพร้อมกับชื่อที่ใส่ ในที่นี้ใส่ชื่อ Score

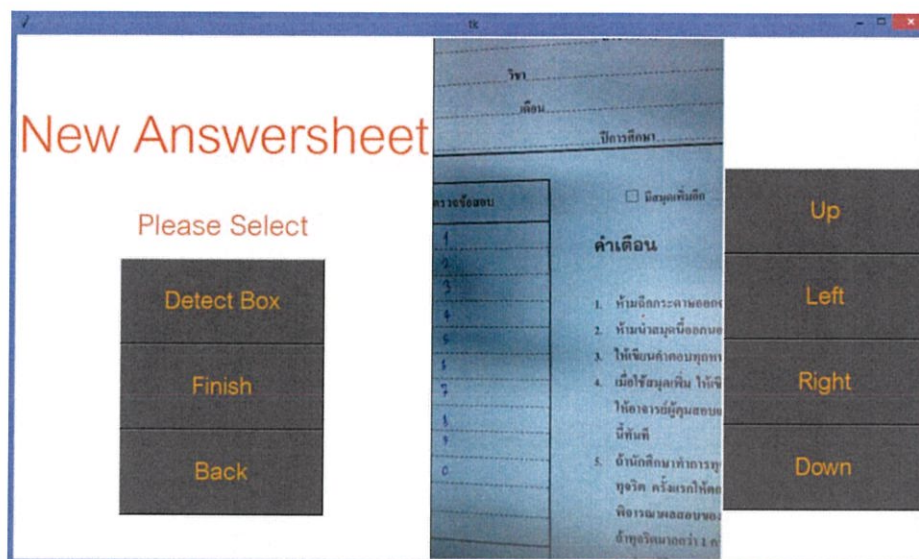


กดปุ่ม Yes



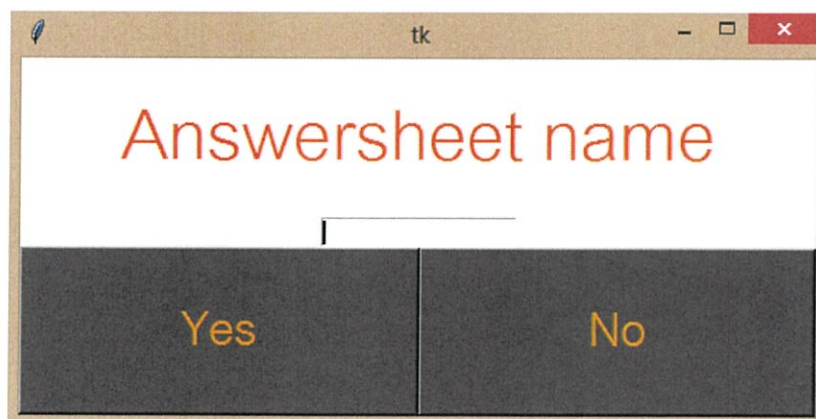
รูปที่ 4.17 ผลที่ได้จากการใส่ชื่อส่วนที่ลากกรอบตรวจจับและกดปุ่ม “Yes”

จากรูปที่ 4.17 ผู้ใช้สามารถลากกรอบตรวจจับเพิ่มเติมได้อีก เมื่อผู้ใช้ไม่ลากกรอบตรวจจับเพิ่มเติมแล้วกดปุ่ม “Ok” จะไปยังหน้าต่างเลื่อนตัวกล่อง ตัวกล่องจะกลับมาที่ตำแหน่งกึ่งกลางของภาครองชุดข้อสอบ



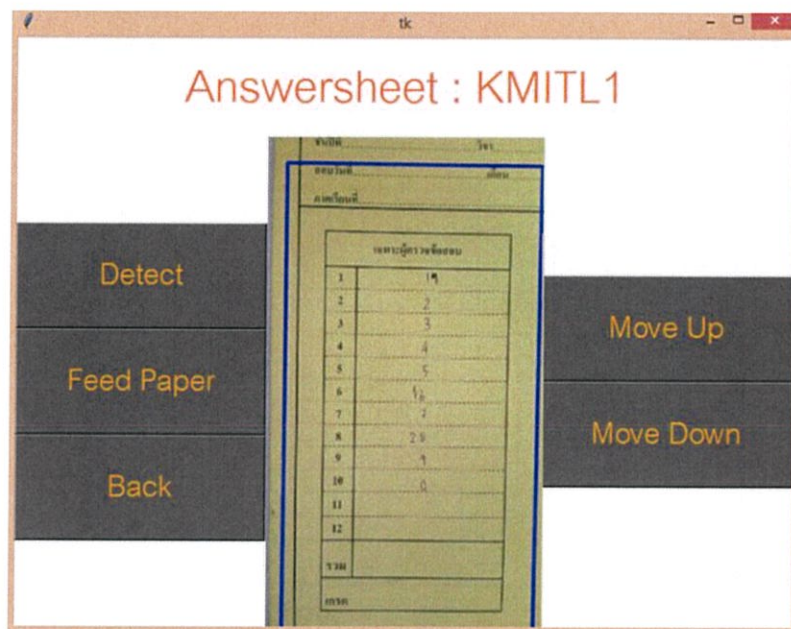
รูปที่ 4.18 ผลที่ได้จากการกดปุ่ม “Ok”

จากรูปที่ 4.19 เมื่อผู้ใช้กดปุ่ม “Finish” จะไปยังหน้ายืนยันชื่อชุดข้อสอบ แสดงได้ดังรูปที่ 4.18



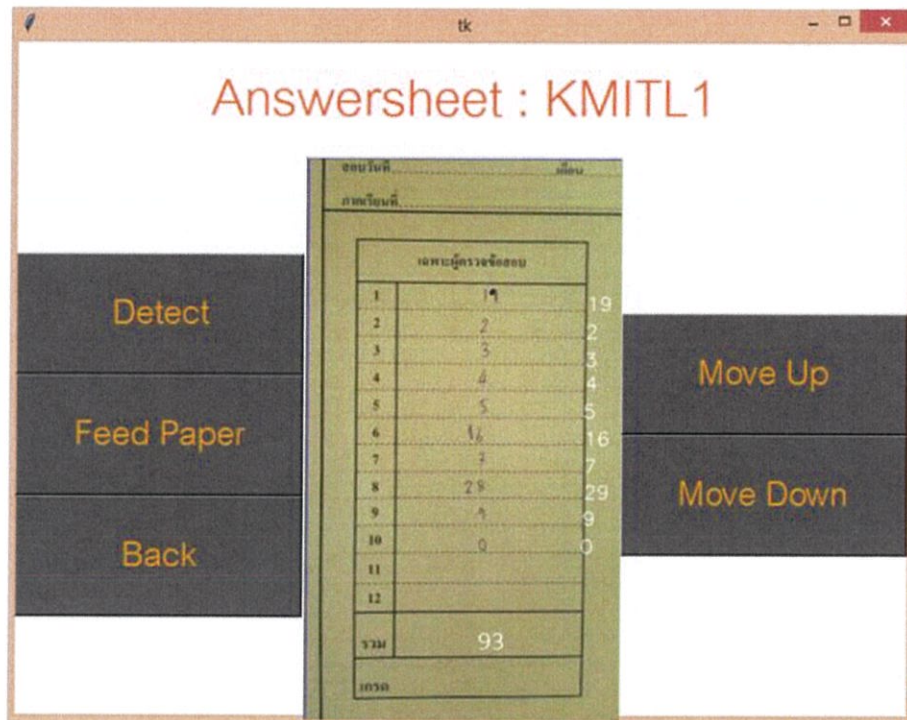
รูปที่ 4.19 ผลที่ได้จากการกดปุ่ม “Finish”

จากรูปที่ 4.19 เมื่อใส่ชีตข้อมูลข้อสอบลงในช่องว่างแล้วกดปุ่ม “Yes” จะไปยังหน้าต่างตรวจจับตัวเลข เมื่อกดปุ่ม “No” จะย้อนกลับไปหน้าต่างเลื่อนตัวกล้อง ในที่นี้ตั้งชื่อชุดข้อสอบว่า KMITL1



รูปที่ 4.20 ผลที่ได้จากการกดปุ่ม “Yes”

จากรูปที่ 4.20 เมื่อกดปุ่ม “Detect” แล้วโปรแกรมจะทำการตรวจจับตัวเลขลายมือ แล้วแสดงผลที่ได้จากการจำแนกตัวเลขในภาพ และนำผลลัพธ์ที่ได้มารวมกันแล้วแสดงผลดังรูปที่ 4.21



รูปที่ 4.21 ผลที่ได้จากการกดปุ่ม “Detect”

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

ปริญญานิพนธ์นี้เป็นการออกแบบและสร้างระบบรวมคะแนนข้อสอบอัตโนมัติพร้อมกับเครื่องเปลี่ยนชุดข้อสอบ โดยแบ่งออกแบบเป็น 3 ส่วน ส่วนแรกคือโปรแกรมเรียนรู้และจดจำตัวเลขลายมือจากการเรียนรู้เชิงลึกโดยใช้ภาษาไพธอนร่วมกับ Library TensorFlow ซึ่งเลือกใช้วิธีการแบบ Convolutional neural network (CNN) เพื่อจำแนกตัวเลข 0 ถึง 9 ซึ่งมีความแม่นยำร้อยละ 87.8 และสามารถจำแนกตัวเลขได้มากกว่า 1 หลัก ส่วนที่ 2 คือโปรแกรมค้นหาตำแหน่งของตัวเลขโดยใช้ภาษาไพธอนร่วมกับ Library OpenCV เพื่อตัดเอาเฉพาะส่วนที่มีตัวเลขก่อนเข้ากระบวนการเรียนรู้เชิงลึกเพื่อจำแนกตัวเลข และผู้ใช้สามารถกำหนดบริเวณการค้นหาตัวเลขจากลายมือได้ เมื่อนำไปใช้กับกระดาษคำตอบรูปแบบอื่นๆ และส่วนที่ 3 เครื่องเปลี่ยนชุดข้อสอบโดยใช้ชุดไมโครคอนโทรลเลอร์ร่วมกับอุปกรณ์มอเตอร์ขับเคลื่อนสายพาน ที่ควบคุมกลไกเลื่อนชุดข้อสอบ ซึ่งมีความผิดพลาดในการทำงานร้อยละ 19.75 และการเลื่อนตำแหน่งกล่องที่ใช้ในการจับภาพซึ่งมีความคลาดเคลื่อน 0.0579 เซนติเมตร

5.2 ข้อเสนอแนะ

5.2.1 ในส่วนการทำงานของชุดเคลื่อนตัวกล่องสองแกน อาจจะสร้างวงจกรองสัญญาณรบกวนจากกระแสไฟฟ้าที่เข้าไปยัง Stepper motor

5.2.2 ในส่วนโปรแกรมประมวลผลภาพ การตัดกรอบตัวเลขยังมีความผิดพลาด อาจพิจารณาใช้วิธีการอื่นแทนวิธี Find Contours

5.2.3 ในส่วนเครื่องเปลี่ยนชุดข้อสอบ ชุดยกลาดรองชุดข้อสอบควรติดตั้งอุปกรณ์ตรวจวัดระยะชุดข้อสอบเมื่อถึงระยะเลื่อนชุดข้อสอบออก และปรับปรุงการทำงานของชุดเลื่อนชุดข้อสอบให้มีความแม่นยำมากขึ้น

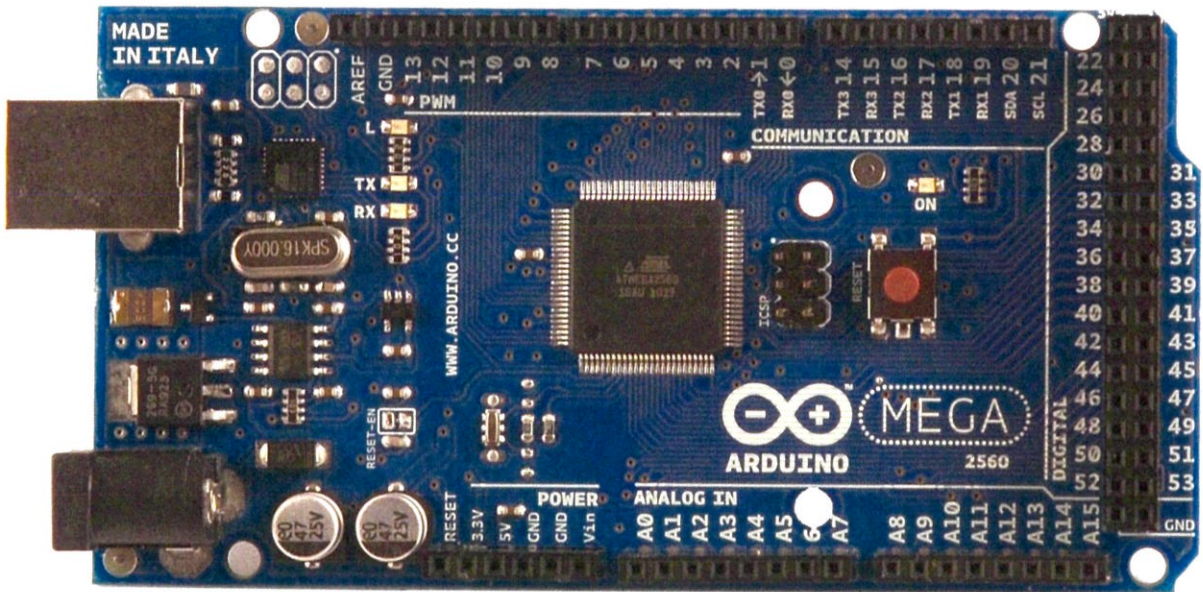
บรรณานุกรม

- [1] montri.rmutl.ac.th. “มอเตอร์ไฟฟ้ากระแสตรง.”
<http://montri.rmutl.ac.th/assets/dc06.pdf>
- [2] zhengkemotor.com. “ZGA25RP/ZGB25RP.”
<http://www.zhengkemotor.com/product/product20.html>
- [3] George. “Unipolar Stepper Motor vs Bipolar Stepper Motors”
<https://www.circuitspecialists.com/blog/unipolar-stepper-motor-vs-bipolar-stepper-motors/>
- [4] Adisak chinawong. “สเตปป์มอเตอร์ (Stepping Motor).”
<http://www.adisak51.com/page22.html>
- [5] arduitrronics.com. “Ultrasonic Sensor Module (HC-SR04).”
<https://www.arduitronics.com/product/20/ultrasonic-sensor-module-hc-sr04>
- [6] วิกิพีเดีย. “ระบบพิกัดคาร์ทีเซียน.”
<https://th.wikipedia.org/wiki/ระบบพิกัดคาร์ทีเซียน>
- [7] Ilan E. Moyer. “Principle of Operation.”
<http://corexy.com/theory.html>
- [8] Nikhil Buduma, Nicholas Lacascio. Fundamentals of Deep Learning. 1sted.
Sebastopol, CA: O’Reilly Media, 2017
- [9] SAURABHYADAV. “FAMOUS DEEP CONVOLUTIONAL NEURAL NETWORKS IN IMAGE PROCESSING.”
<http://themlguy.com/index.php/2016/11/16/overview-famous-deep-convolutional-neural-networks-in-image-processing/>
- [10] kenmatsu. “Convolutional Neural Network.”
<https://www.slideshare.net/matsukenbook/deep-learning-chap6-convolutional-neural-net>

- [11] aliostad. "Future of Programming - Rise of the Scientific Programmer."
<http://byterot.blogspot.com/2015/01/future-of-programming-rise-of-the-scientific-developer-bigdata-datascience-machine-learning-and-fall-of-the-craftsman.html>
- [12] commandronestore. "A4988 Stepper Driver."
<http://commandronestore.com/products/cm4988.php>
- [13] arduinoall.com. "3D Printer Endstop Switch RAMPS 1.4 Mechanical Limit Switches."
<https://www.arduinoall.com/product/1788/3d-printer-endstop-switch-ramps-1-4-mechanical-limit-switches>
- [14] ThaiEasyElec.com. "Arduino Mega 2560."
<http://www.thaieasyelec.com/products/development-boards/arduino/official-boards-made-in-italy/arduino-mega-2560-detail.html>
- [15] arduinoall.com. "L298N ไอซีขับเคลื่อนมอเตอร์ L298N Full-Bridge Dual Motor Driver L298 IC."
<https://www.arduinoall.com/product/571/l298n-ไอซีขับเคลื่อนมอเตอร์-l298n-full-bridge-dual-motor-driver-l298-ic>
- [16] Banggood Ltd. (2006). "JKM NEMA17 42 Hybrid Stepper Motor 2 Phase 1.8 Degree 40MM for CNC Router."
https://www.banggood.com/JKM-42-Hybrid-Stepper-Motor-2-Phase-1_840MM-For-CNC-Router-p-933548.html?p=9X051049165582016063
- [17] "NOTEBOOK ASUS TRANSFORMER BOOK FLIP TP550LD."
<https://www.jib.co.th/web/index.php/product/readProduct/15699/2/Notebook-Asus-Transformer-Book-Flip-Tp550ld-cj078h>
- [18] "WEB CAM FULL HD 386."
http://nabees.ran4u.com/p_10746_50835_447745_WEB-CAM-FULL-HD-386.htm

ภาคผนวก

Arduino MEGA 2560



Product Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Index

Technical Specifications

Page 2

How to use Arduino
Programming Environment, Basic Tutorials

Page 6

Terms & Conditions

Page 7

Environmental Policies
half sqm of green via Impatto Zero®

Page 7



radiospares

RADIONICS



Technical Specification

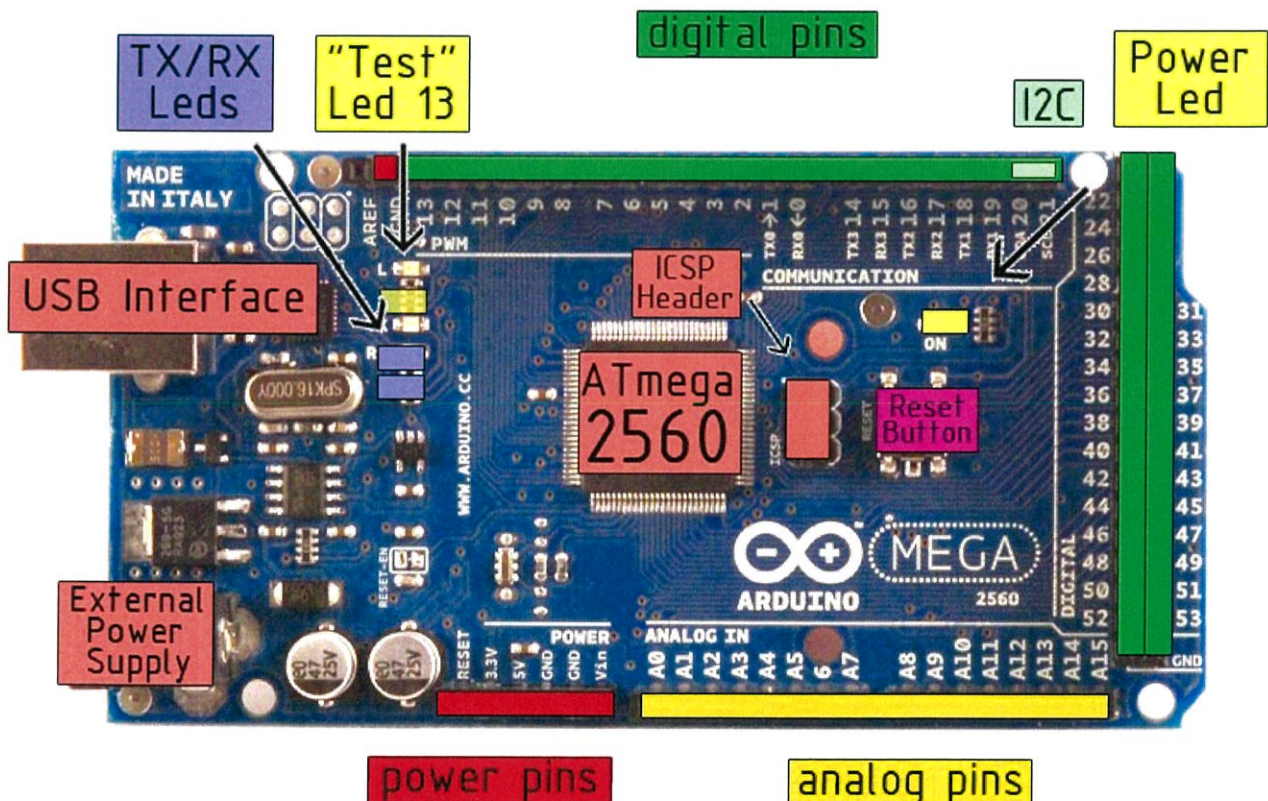


EAGLE files: [arduino-mega2560-reference-design.zip](#) Schematic: [arduino-mega2560-schematic.pdf](#)

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

the board



radiospares

RADIONICS



Power

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I²C: 20 (SDA) and 21 (SCL).** Support I²C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I²C pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and [analogReference\(\)](#) function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.



radiospares

RADIONICS



The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega's digital pins.

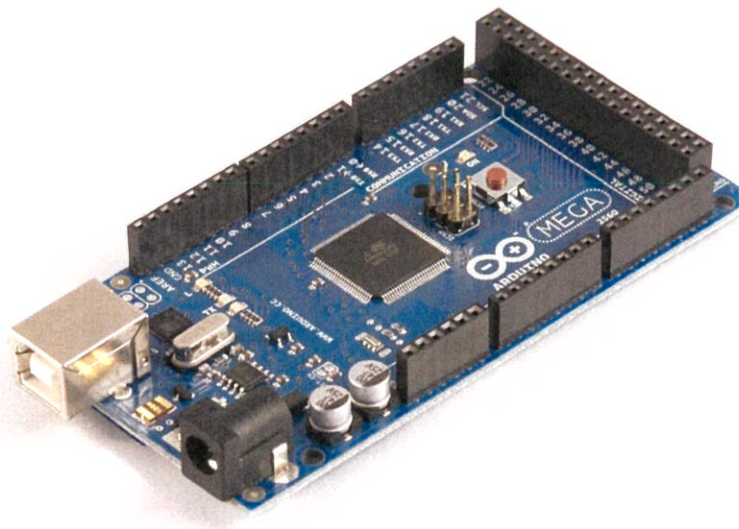
The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation on the Wiring website](#) for details. To use the SPI communication, please see the ATmega2560 datasheet.

Programming

The Arduino Mega2560 can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.



radiospares

RADIONICS



Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Mega has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila. **Please note that I²C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).**



radiospares

RADIONICS



How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink**

Once you have your sketch you'll see something very close to the screenshot on the right.

In **Tools>Board** select MEGA

Now you have to go to **Tools>SerialPort** and select the right serial port, the one arduino is attached to.

```
Blink | Arduino 0017
File Edit Sketch Tools Help
Blink $
int ledPin = 13; // LED connected to digital pin 13
// The setup() method runs once, when the sketch starts
void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}
// the loop() method runs over and over again,
// as long as the Arduino has power
void loop()
{
  digitalWrite(ledPin, HIGH); // set the LED on
  delay(1000); // wait for a second
  digitalWrite(ledPin, LOW); // set the LED off
  delay(1000); // wait for a second
}
```



Done compiling.

Press Compile button
(to check for errors)



Upload



TX RX Flashing



Blinking Led!

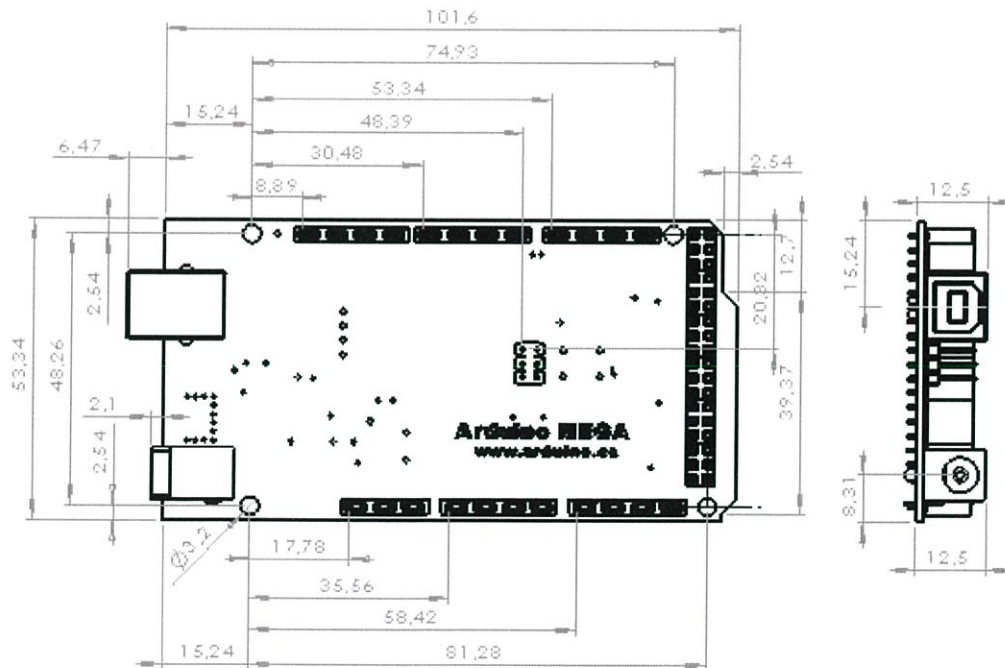
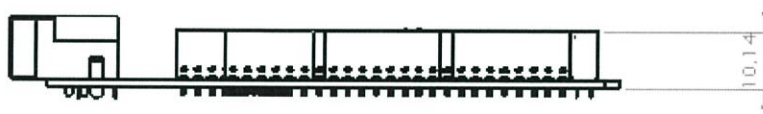
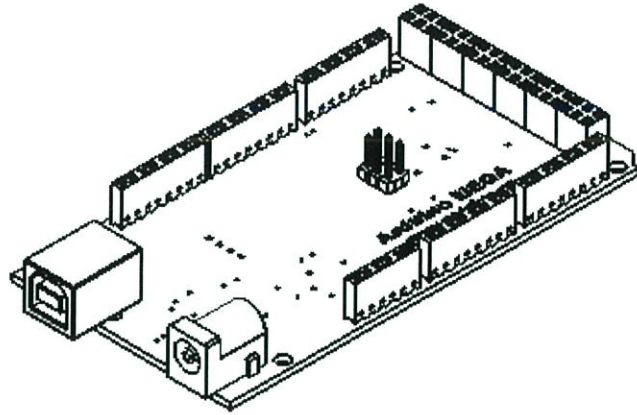


radiospares

RADIONICS



Dimensioned Drawing



radiospares **RADIONICS**



Terms & Conditions



1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.



Environmental Policies



The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.



radiospares

RADIONICS

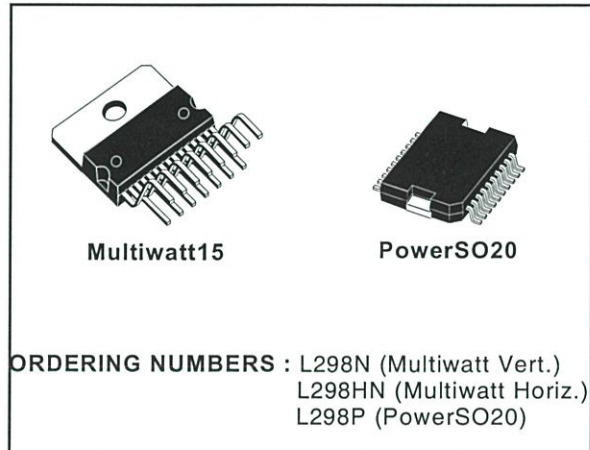


DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

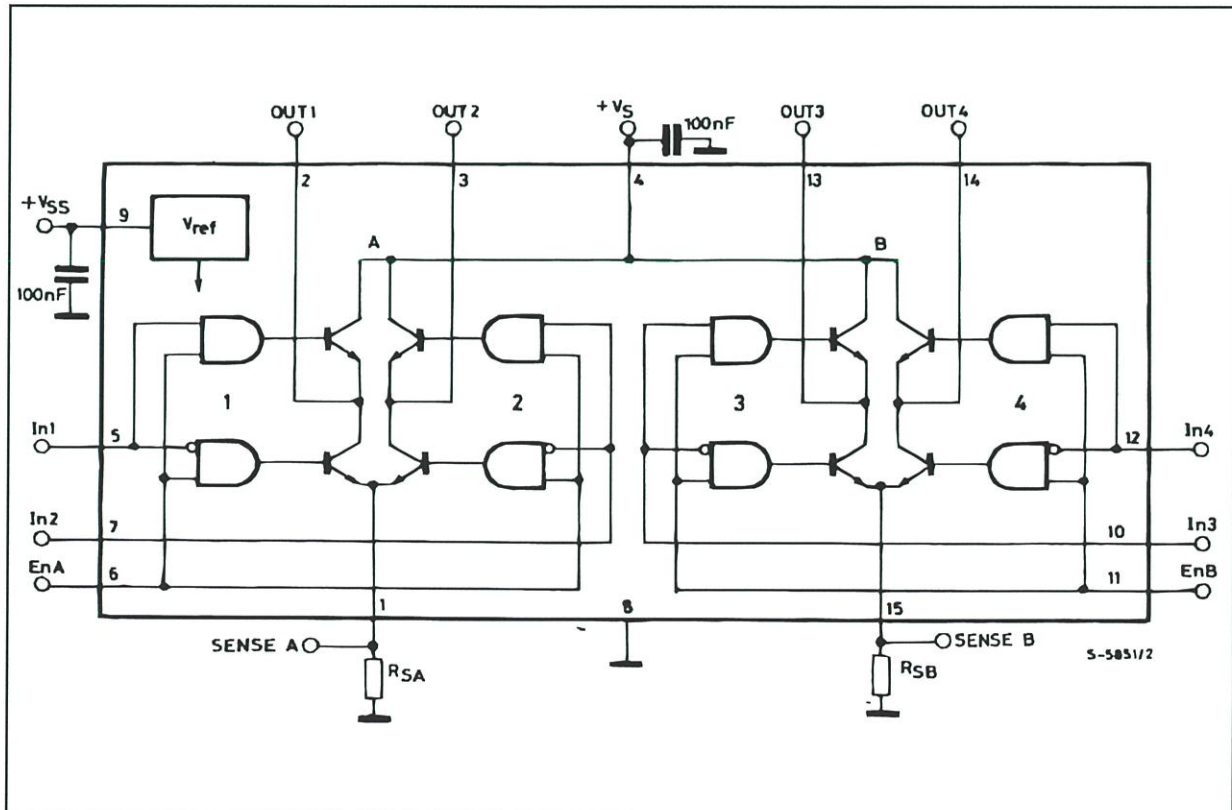
DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

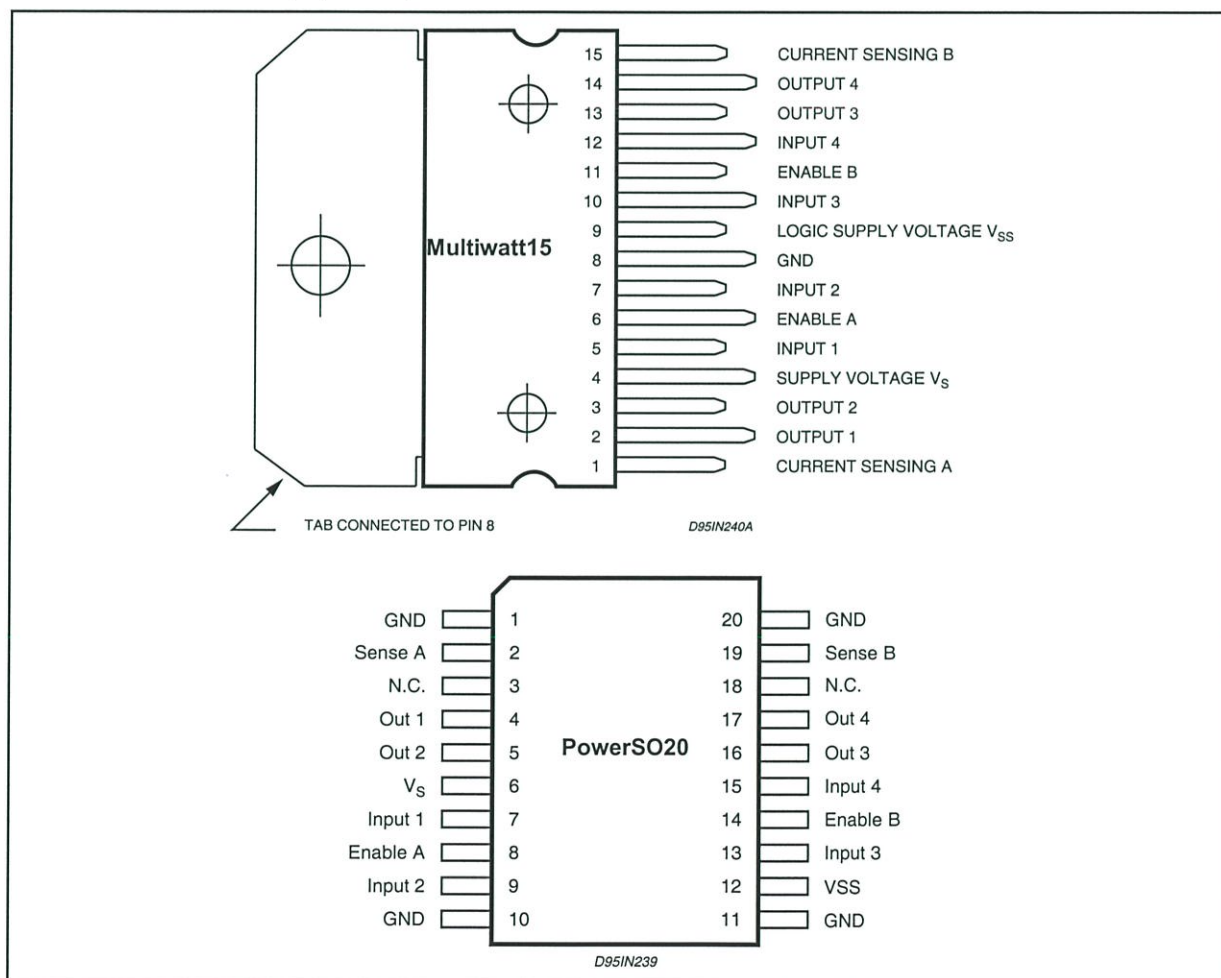
BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_I, V_{en}	Input and Enable Voltage	-0.3 to 7	V
I_O	Peak Output Current (each Channel)		
	- Non Repetitive (t = 100 μ s)	3	A
	- Repetitive (80% on -20% off; t _{on} = 10ms)	2.5	A
	-DC Operation	2	A
V_{sens}	Sensing Voltage	-1 to 2.3	V
P_{tot}	Total Power Dissipation (T _{case} = 75°C)	25	W
T_{op}	Junction Operating Temperature	-25 to 130	°C
T_{stg}, T_j	Storage and Junction Temperature	-40 to 150	°C

PIN CONNECTIONS (top view)



THERMAL DATA

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{th\ j-case}$	Thermal Resistance Junction-case	Max.	-	3	°C/W
$R_{th\ j-amb}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	°C/W

(*) Mounted on aluminum substrate

PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V _S	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	V _{SS}	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
–	3;18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS (V_S = 42V; V_{SS} = 5V, T_j = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V _S	Supply Voltage (pin 4)	Operative Condition	V _{IH} +2.5		46	V
V _{SS}	Logic Supply Voltage (pin 9)		4.5	5	7	V
I _S	Quiescent Supply Current (pin 4)	V _{en} = H; I _L = 0 V _i = L V _i = H		13 50	22 70	mA mA
		V _{en} = L V _i = X			4	mA
I _{SS}	Quiescent Current from V _{SS} (pin 9)	V _{en} = H; I _L = 0 V _i = L V _i = H		24 7	36 12	mA mA
		V _{en} = L V _i = X			6	mA
V _{IL}	Input Low Voltage (pins 5, 7, 10, 12)		–0.3		1.5	V
V _{IH}	Input High Voltage (pins 5, 7, 10, 12)		2.3		V _{SS}	V
I _{IL}	Low Voltage Input Current (pins 5, 7, 10, 12)	V _i = L			–10	μA
I _{IH}	High Voltage Input Current (pins 5, 7, 10, 12)	V _i = H ≤ V _{SS} –0.6V		30	100	μA
V _{en} = L	Enable Low Voltage (pins 6, 11)		–0.3		1.5	V
V _{en} = H	Enable High Voltage (pins 6, 11)		2.3		V _{SS}	V
I _{en} = L	Low Voltage Enable Current (pins 6, 11)	V _{en} = L			–10	μA
I _{en} = H	High Voltage Enable Current (pins 6, 11)	V _{en} = H ≤ V _{SS} –0.6V		30	100	μA
V _{CEsat} (H)	Source Saturation Voltage	I _L = 1A I _L = 2A	0.95	1.35 2	1.7 2.7	V V
V _{CEsat} (L)	Sink Saturation Voltage	I _L = 1A (5) I _L = 2A (5)	0.85	1.2 1.7	1.6 2.3	V V
V _{CEsat}	Total Drop	I _L = 1A (5) I _L = 2A (5)	1.80		3.2 4.9	V V
V _{sens}	Sensing Voltage (pins 1, 15)		–1 (1)		2	V

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
T ₁ (V _i)	Source Current Turn-off Delay	0.5 V _i to 0.9 I _L (2); (4)		1.5		μs
T ₂ (V _i)	Source Current Fall Time	0.9 I _L to 0.1 I _L (2); (4)		0.2		μs
T ₃ (V _i)	Source Current Turn-on Delay	0.5 V _i to 0.1 I _L (2); (4)		2		μs
T ₄ (V _i)	Source Current Rise Time	0.1 I _L to 0.9 I _L (2); (4)		0.7		μs
T ₅ (V _i)	Sink Current Turn-off Delay	0.5 V _i to 0.9 I _L (3); (4)		0.7		μs
T ₆ (V _i)	Sink Current Fall Time	0.9 I _L to 0.1 I _L (3); (4)		0.25		μs
T ₇ (V _i)	Sink Current Turn-on Delay	0.5 V _i to 0.9 I _L (3); (4)		1.6		μs
T ₈ (V _i)	Sink Current Rise Time	0.1 I _L to 0.9 I _L (3); (4)		0.2		μs
f _c (V _i)	Commutation Frequency	I _L = 2A		25	40	KHz
T ₁ (V _{en})	Source Current Turn-off Delay	0.5 V _{en} to 0.9 I _L (2); (4)		3		μs
T ₂ (V _{en})	Source Current Fall Time	0.9 I _L to 0.1 I _L (2); (4)		1		μs
T ₃ (V _{en})	Source Current Turn-on Delay	0.5 V _{en} to 0.1 I _L (2); (4)		0.3		μs
T ₄ (V _{en})	Source Current Rise Time	0.1 I _L to 0.9 I _L (2); (4)		0.4		μs
T ₅ (V _{en})	Sink Current Turn-off Delay	0.5 V _{en} to 0.9 I _L (3); (4)		2.2		μs
T ₆ (V _{en})	Sink Current Fall Time	0.9 I _L to 0.1 I _L (3); (4)		0.35		μs
T ₇ (V _{en})	Sink Current Turn-on Delay	0.5 V _{en} to 0.9 I _L (3); (4)		0.25		μs
T ₈ (V _{en})	Sink Current Rise Time	0.1 I _L to 0.9 I _L (3); (4)		0.1		μs

- 1) Sensing voltage can be -1 V for t ≤ 50 μsec; in steady state V_{sens} min ≥ -0.5 V.
- 2) See fig. 2.
- 3) See fig. 4.
- 4) The load must be a pure resistor.

Figure 1 : Typical Saturation Voltage vs. Output Current.

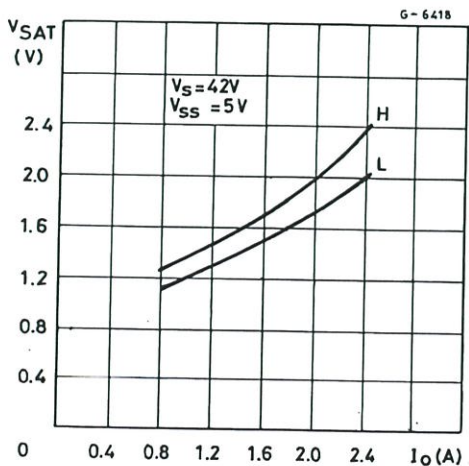
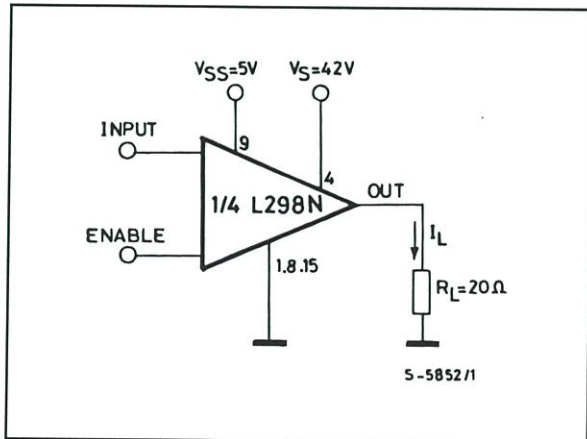


Figure 2 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
 For ENABLE Switching, set IN = H

Figure 3 : Source Current Delay Times vs. Input or Enable Switching.

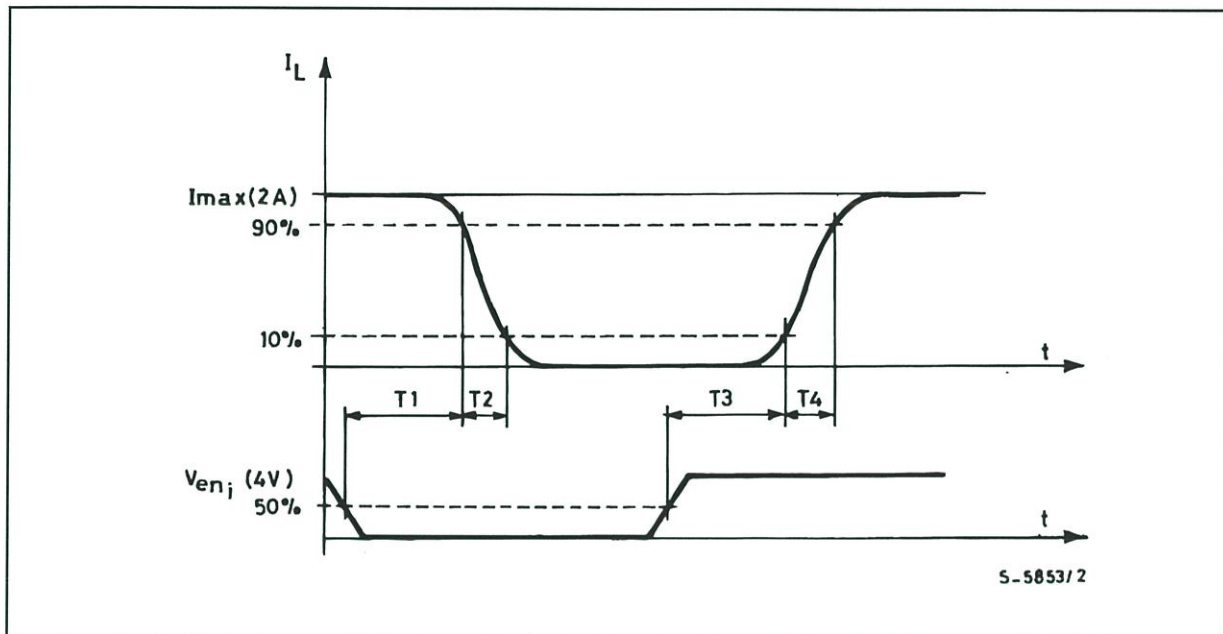
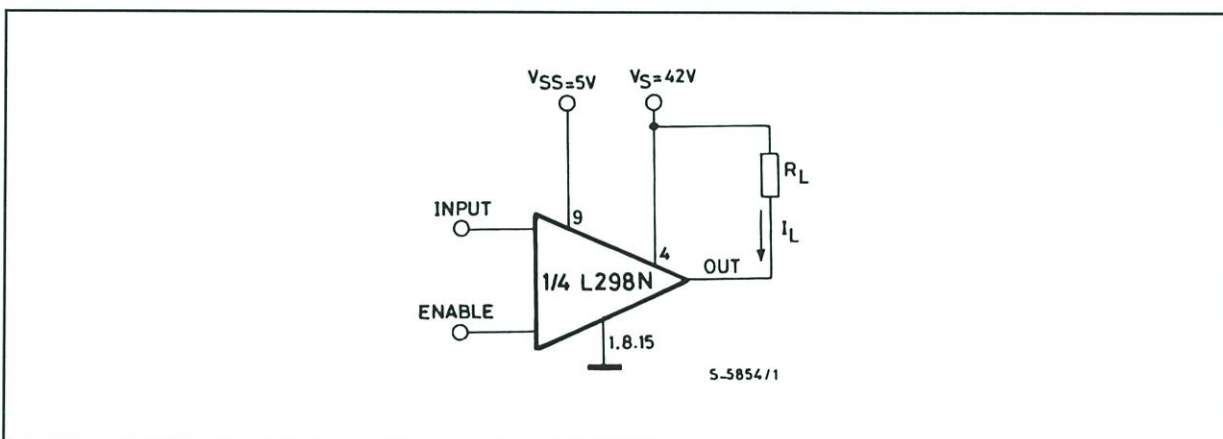


Figure 4 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
 For ENABLE Switching, set IN = L

Figure 5 : Sink Current Delay Times vs. Input 0 V Enable Switching.

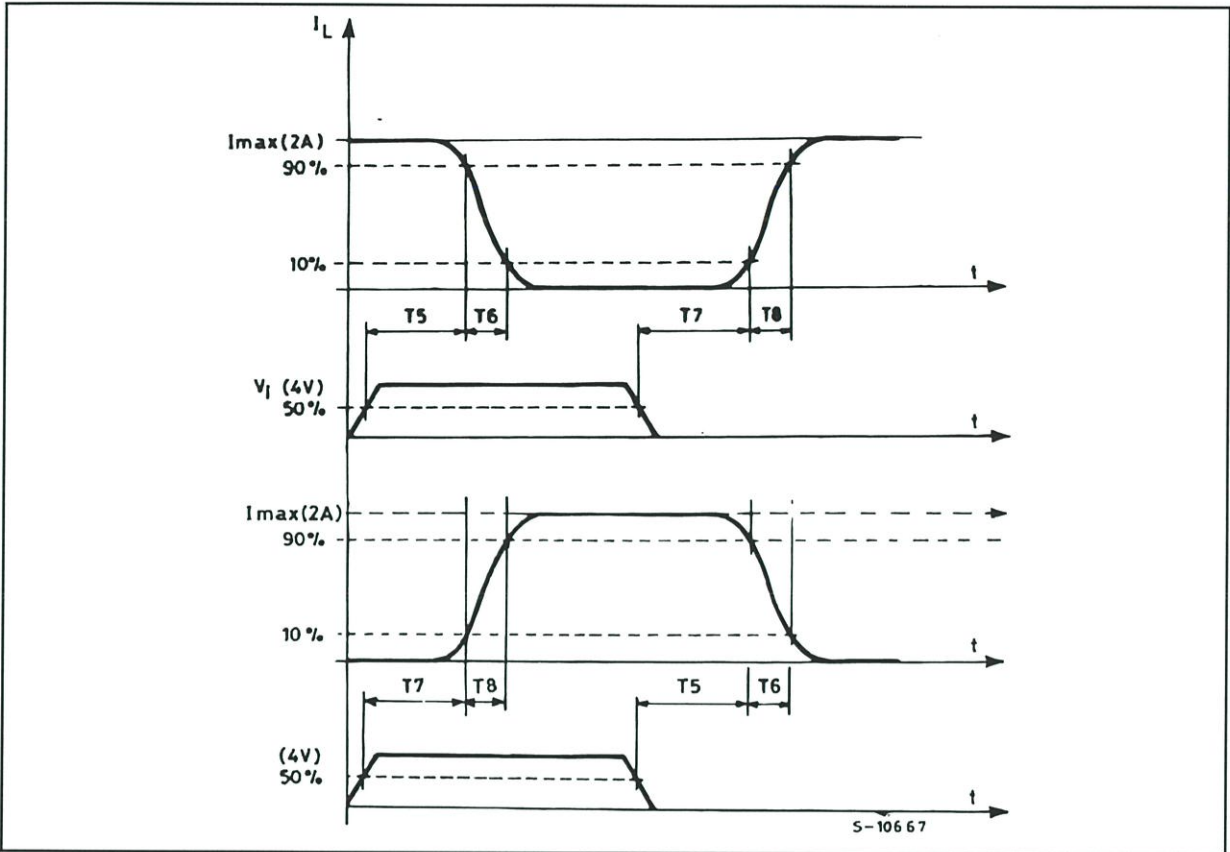


Figure 6 : Bidirectional DC Motor Control.

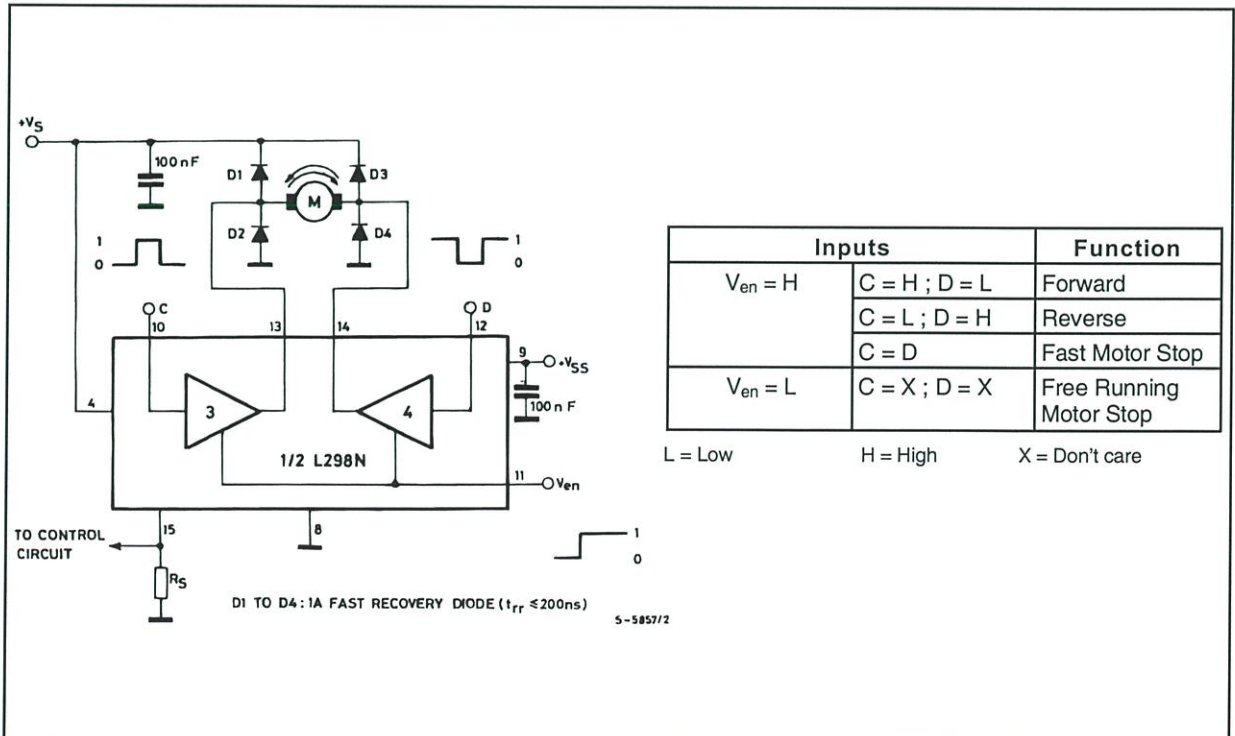
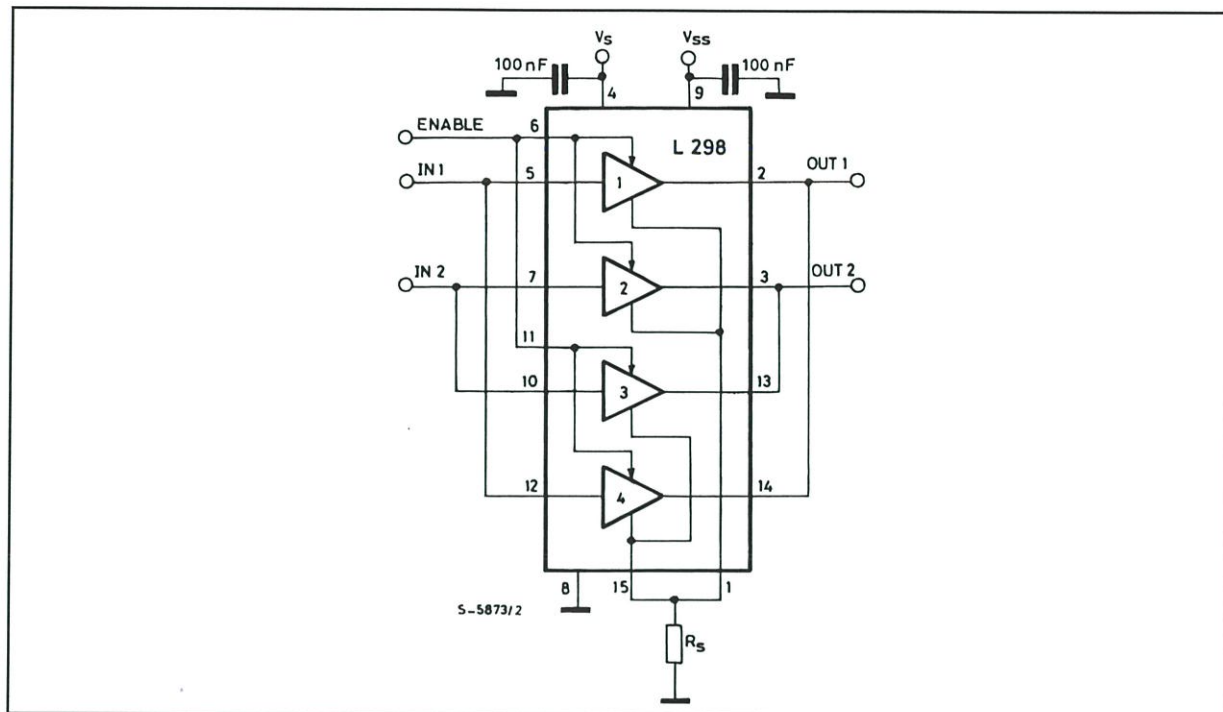


Figure 7 : For higher currents, outputs can be paralleled. Take care to parallel channel 1 with channel 4 and channel 2 with channel 3.



APPLICATION INFORMATION (Refer to the block diagram)

1.1. POWER OUTPUT STAGE

The L298 integrates two power output stages (A ; B). The power output stage is a bridge configuration and its outputs can drive an inductive load in common or differential mode, depending on the state of the inputs. The current that flows through the load comes out from the bridge at the sense output : an external resistor (R_{SA} ; R_{SB} .) allows to detect the intensity of this current.

1.2. INPUT STAGE

Each bridge is driven by means of four gates the input of which are $In1$; $In2$; EnA and $In3$; $In4$; EnB . The In inputs set the bridge state when The En input is high ; a low state of the En input inhibits the bridge. All the inputs are TTL compatible.

2. SUGGESTIONS

A non inductive capacitor, usually of 100 nF, must be foreseen between both V_s and V_{ss} , to ground, as near as possible to GND pin. When the large capacitor of the power supply is too far from the IC, a second smaller one must be foreseen near the L298.

The sense resistor, not of a wire wound type, must be grounded near the negative pole of V_s that must be near the GND pin of the I.C.

Each input must be connected to the source of the driving signals by means of a very short path.

Turn-On and Turn-Off : Before to Turn-ON the Supply Voltage and before to Turn it OFF, the Enable input must be driven to the Low state.

3. APPLICATIONS

Fig 6 shows a bidirectional DC motor control Schematic Diagram for which only one bridge is needed. The external bridge of diodes $D1$ to $D4$ is made by four fast recovery elements ($t_{rr} \leq 200$ nsec) that must be chosen of a VF as low as possible at the worst case of the load current.

The sense output voltage can be used to control the current amplitude by chopping the inputs, or to provide overcurrent protection by switching low the enable input.

The brake function (Fast motor stop) requires that the Absolute Maximum Rating of 2 Amps must never be overcome.

When the repetitive peak current needed from the load is higher than 2 Amps, a paralleled configuration can be chosen (See Fig.7).

An external bridge of diodes are required when inductive loads are driven and when the inputs of the IC are chopped ; Schottky diodes would be preferred.

This solution can drive until 3 Amps In DC operation and until 3.5 Amps of a repetitive peak current.

On Fig 8 it is shown the driving of a two phase bipolar stepper motor ; the needed signals to drive the inputs of the L298 are generated, in this example, from the IC L297.

Fig 9 shows an example of P.C.B. designed for the application of Fig 8.

Figure 8 : Two Phase Bipolar Stepper Motor Circuit.

This circuit drives bipolar stepper motors with winding currents up to 2 A. The diodes are fast 2 A types.

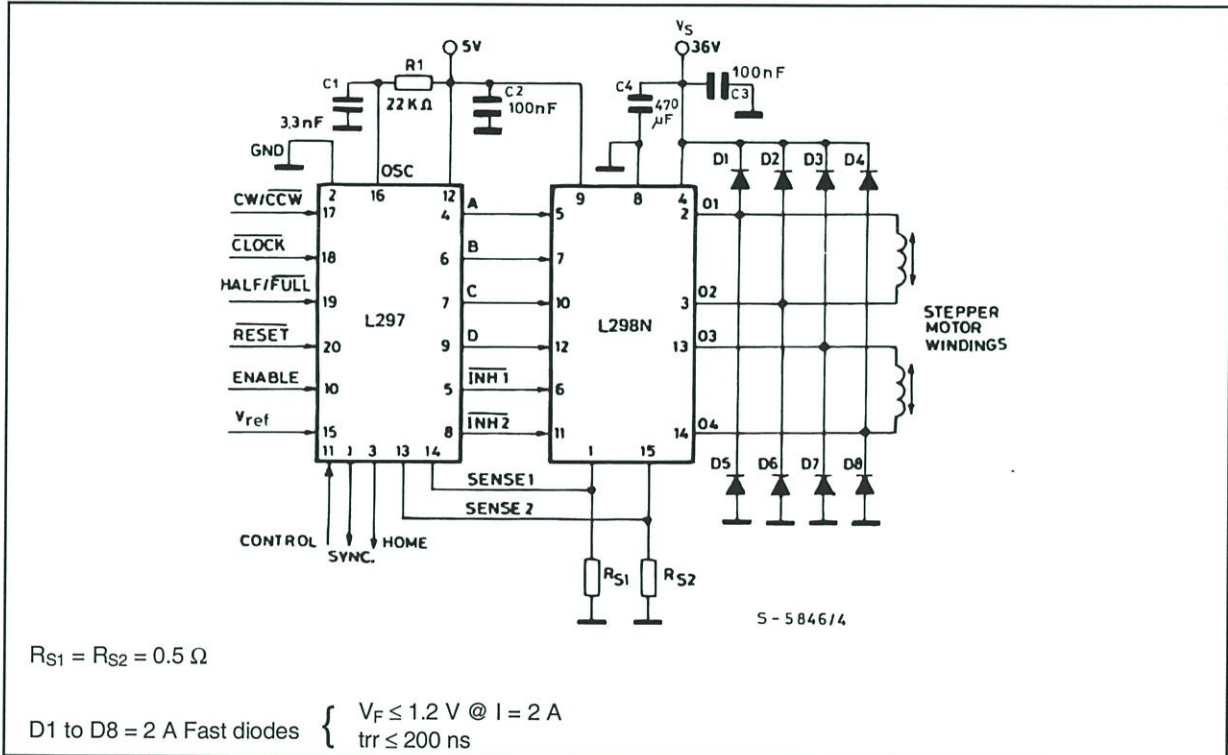


Fig 10 shows a second two phase bipolar stepper motor control circuit where the current is controlled by the I.C. L6506.

Figure 9 : Suggested Printed Circuit Board Layout for the Circuit of fig. 8 (1:1 scale).

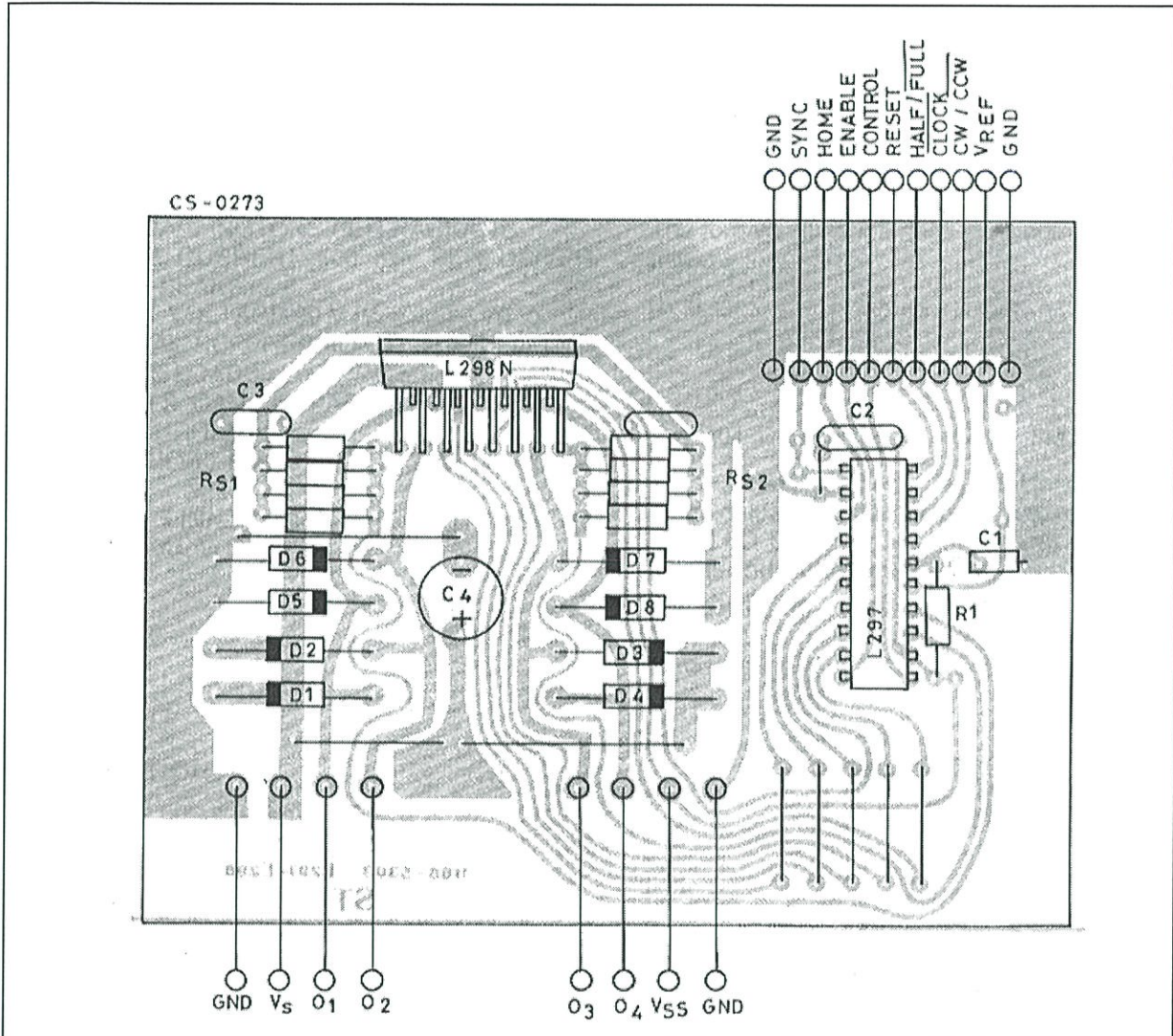
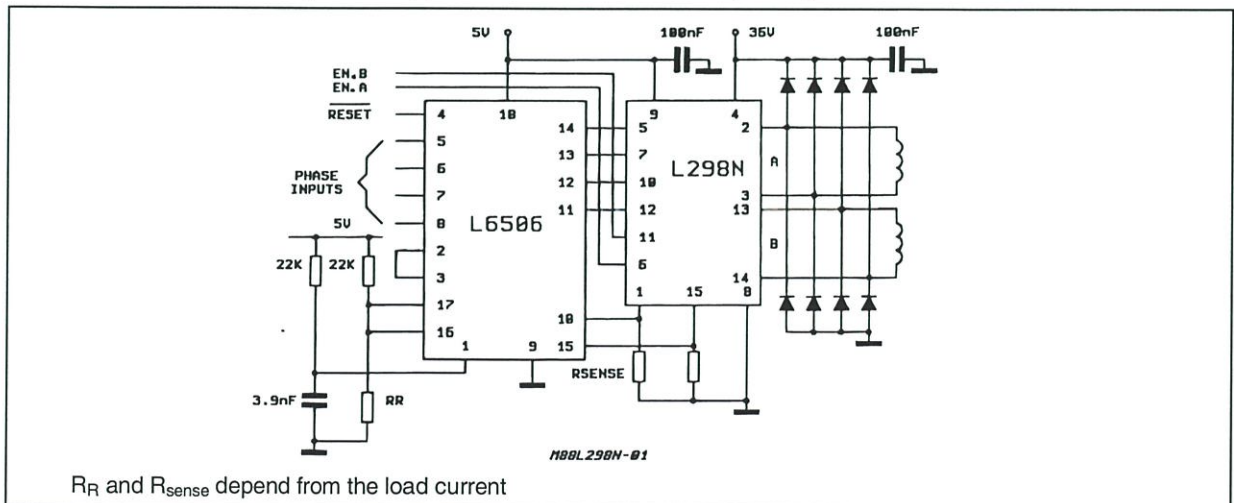
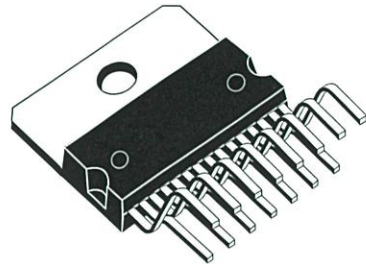


Figure 10 : Two Phase Bipolar Stepper Motor Control Circuit by Using the Current Controller L6506.

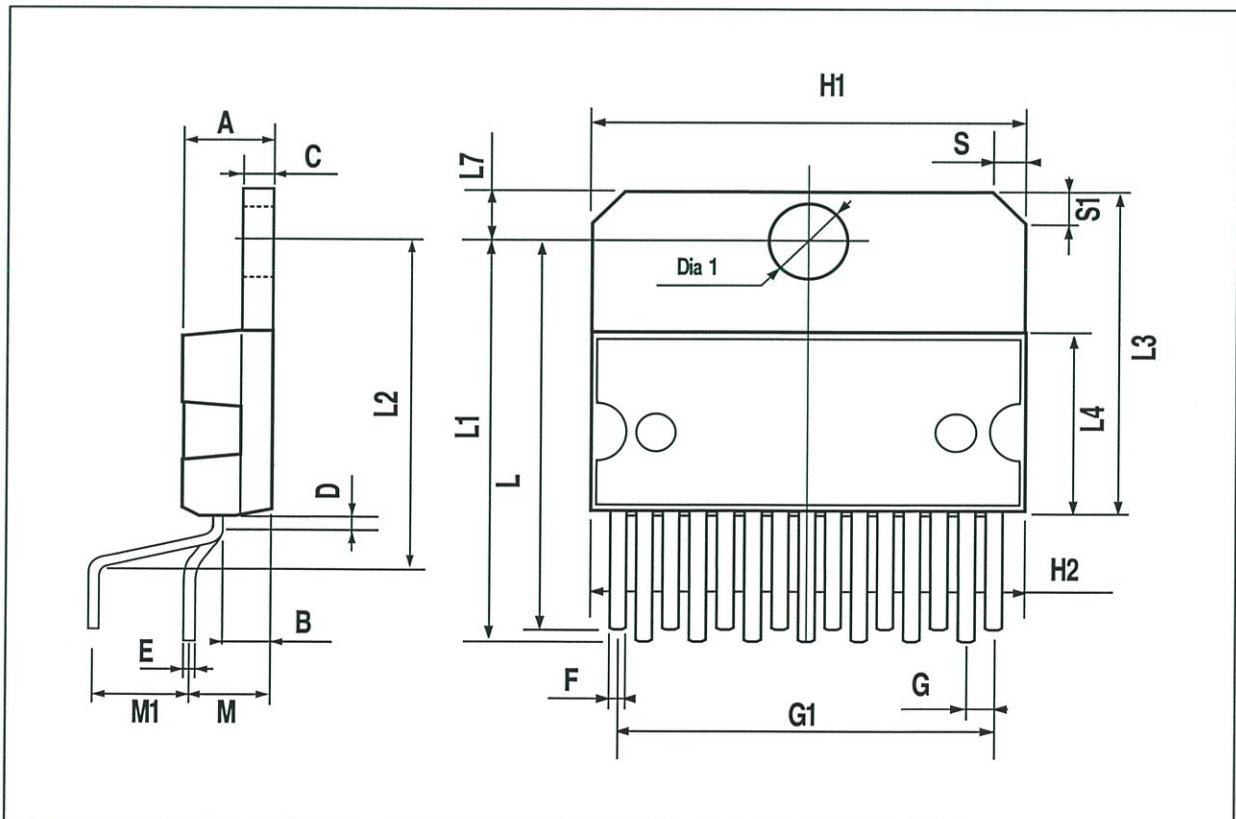


DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
D		1			0.039	
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.02	1.27	1.52	0.040	0.050	0.060
G1	17.53	17.78	18.03	0.690	0.700	0.710
H1	19.6			0.772		
H2			20.2			0.795
L	21.9	22.2	22.5	0.862	0.874	0.886
L1	21.7	22.1	22.5	0.854	0.870	0.886
L2	17.65		18.1	0.695		0.713
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L7	2.65		2.9	0.104		0.114
M	4.25	4.55	4.85	0.167	0.179	0.191
M1	4.63	5.08	5.53	0.182	0.200	0.218
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

OUTLINE AND MECHANICAL DATA

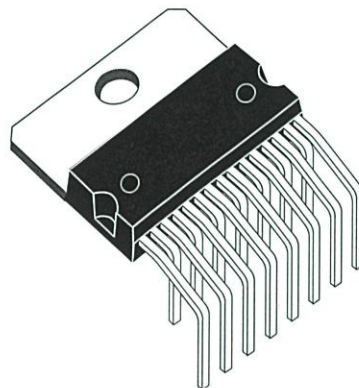


Multiwatt15 V

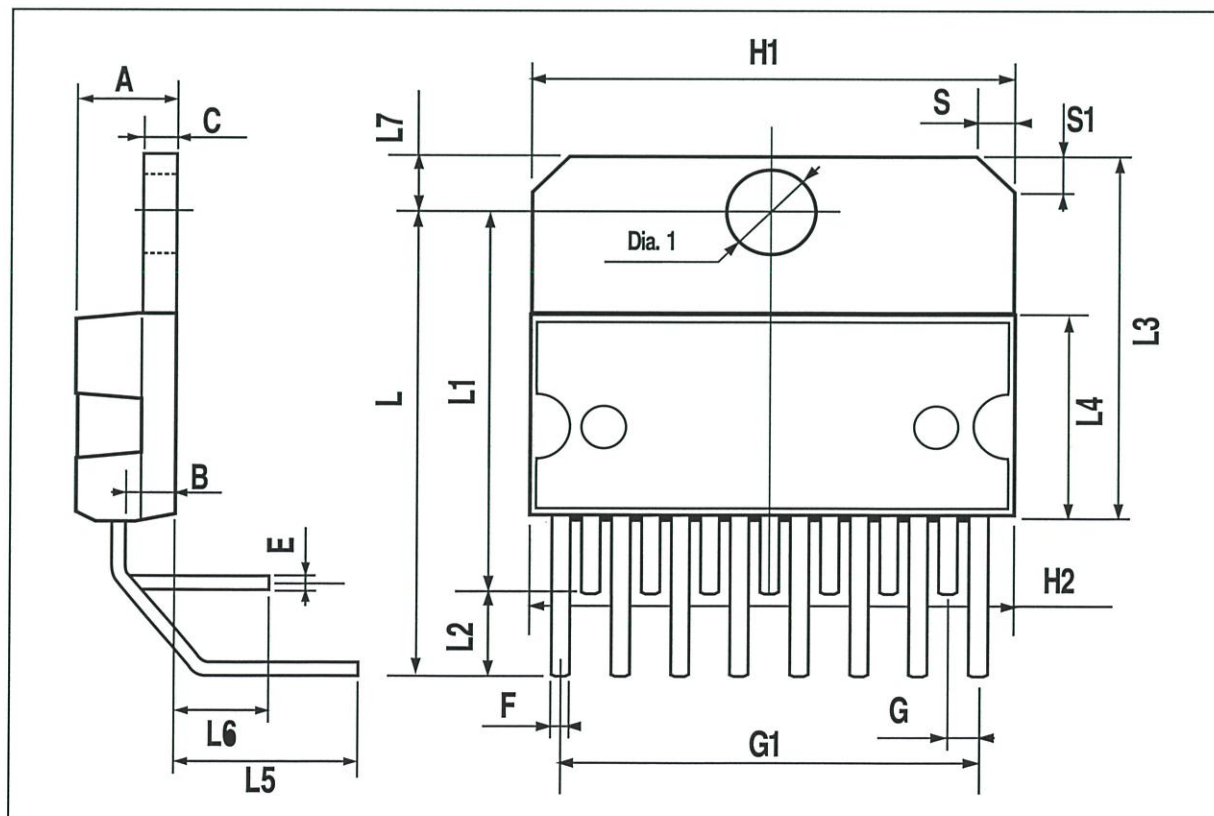


DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.14	1.27	1.4	0.045	0.050	0.055
G1	17.57	17.78	17.91	0.692	0.700	0.705
H1	19.6			0.772		
H2			20.2			0.795
L		20.57			0.810	
L1		18.03			0.710	
L2		2.54			0.100	
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L5		5.28			0.208	
L6		2.38			0.094	
L7	2.65		2.9	0.104		0.114
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

OUTLINE AND MECHANICAL DATA



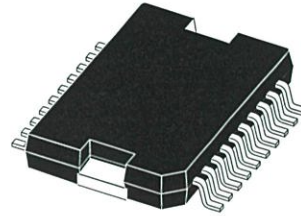
Multiwatt15 H



DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			3.6			0.142
a1	0.1		0.3	0.004		0.012
a2			3.3			0.130
a3	0		0.1	0.000		0.004
b	0.4		0.53	0.016		0.021
c	0.23		0.32	0.009		0.013
D (1)	15.8		16	0.622		0.630
D1	9.4		9.8	0.370		0.386
E	13.9		14.5	0.547		0.570
e		1.27			0.050	
e3		11.43			0.450	
E1 (1)	10.9		11.1	0.429		0.437
E2			2.9			0.114
E3	5.8		6.2	0.228		0.244
G	0		0.1	0.000		0.004
H	15.5		15.9	0.610		0.626
h			1.1			0.043
L	0.8		1.1	0.031		0.043
N	10° (max.)					
S	8° (max.)					
T		10			0.394	

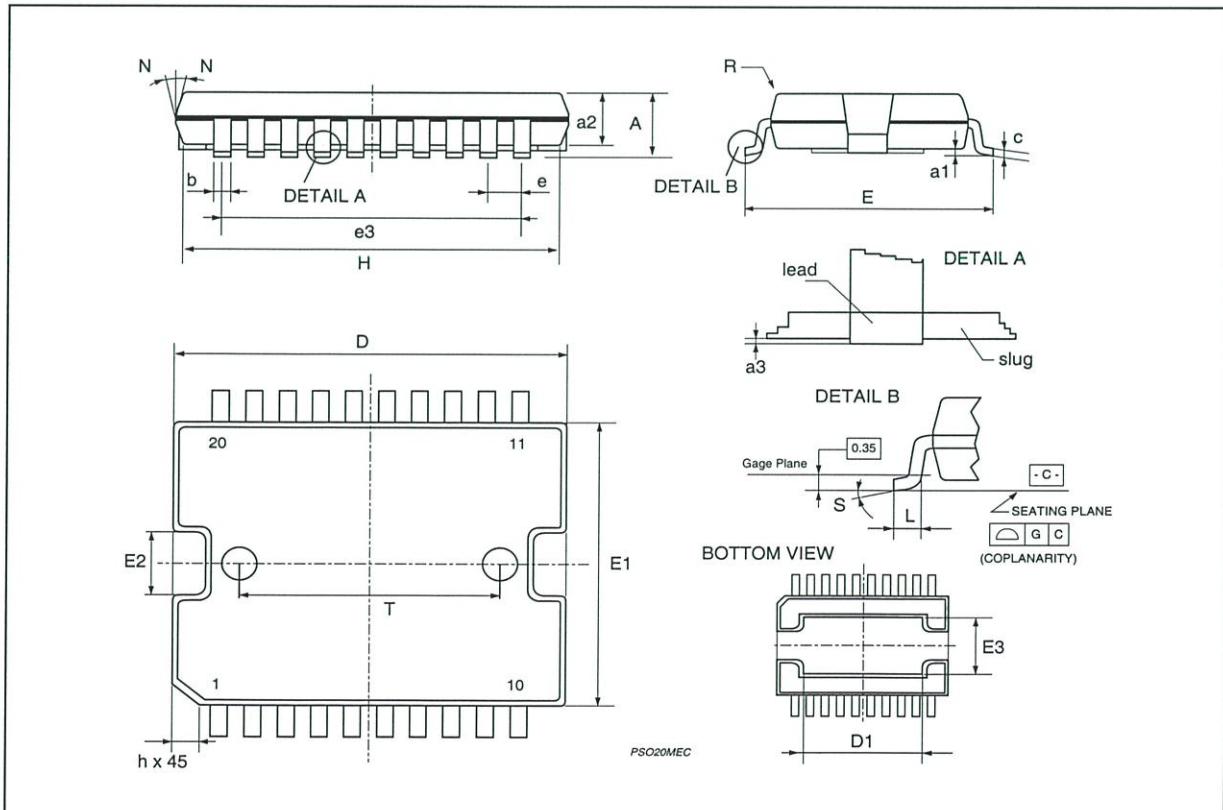
(1) "D and F" do not include mold flash or protrusions.
 - Mold flash or protrusions shall not exceed 0.15 mm (0.006").
 - Critical dimensions: "E", "G" and "a3"

OUTLINE AND MECHANICAL DATA



JEDEC MO-166

PowerSO20

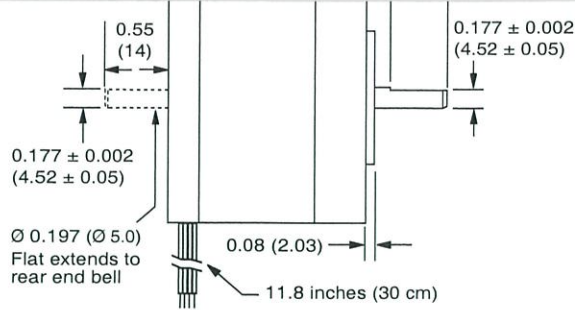


Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specification mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

The ST logo is a registered trademark of STMicroelectronics
© 2000 STMicroelectronics – Printed in Italy – All Rights Reserved
STMicroelectronics GROUP OF COMPANIES

Australia - Brazil - China - Finland - France - Germany - Hong Kong - India - Italy - Japan - Malaysia - Malta - Morocco -
Singapore - Spain - Sweden - Switzerland - United Kingdom - U.S.A.

<http://www.st.com>



Notes and Warnings

Installation, configuration and maintenance must be carried out by qualified technicians only. You must have detailed information to be able to carry out this work.

- Unexpected dangers may be encountered when working with this product!
- Incorrect use may destroy this product and connected components!

For more information, go to www.imshome.com

Specifications

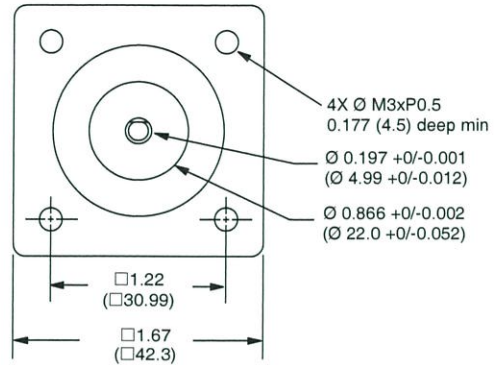
1.5 Amp motors		Single length	Double length	Triple length
Part number		M-1713-1.5 • (1)	M-1715-1.5 • (1)	M-1719-1.5 • (1)
Holding torque	oz-in	32	60	75
	N-cm	23	42	53
Detent torque	oz-in	1.7	2.1	3.5
	N-cm	1.2	1.5	2.5
Rotor inertia	oz-in-sec ²	0.000538	0.0008037	0.0011562
	kg-cm ²	0.038	0.057	0.082
Weight	oz	7.4	8.1	12.7
	grams	210	230	360
Phase current	amps	1.5	1.5	1.5
Phase resistance	ohms	1.3	2.1	2.0
Phase inductance	mH	2.1	5.0	3.85

(1) Indicate S for single-shaft or D for double-shaft. Example M-1713-1.5S

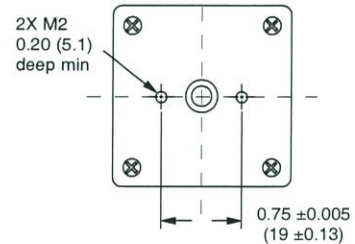
Wiring and Connections

Signals and wire colors	
Phase A	Red
Phase /A	Blue
Phase B	Green
Phase /B	Black

FRONT VIEW



REAR VIEW (Reduced)



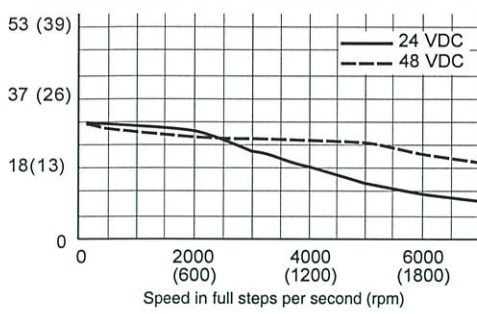
Motor stack length inches (mm)	Single	Double	Triple
LMAX	1.34 (34.0)	1.57 (40)	1.89 (48)

Part Numbers

Example:	M - 1 7 1 3 - 1.5 S
Stepper motor frame size M-17 = NEMA 17 (1.7"/42 mm)	M - 1 7 1 3 - 1.5 S
Motor length 13- = single stack 15- = double stack 19- = triple stack	M - 1 7 1 3 - 1.5 S
Phase current 1.5 = 1.5 Amps	M - 1 7 1 3 - 1.5 S
Shaft S = single, front shaft only D = double, front and rear shafts	M - 1 7 1 3 - 1.5 S
Optional optical encoder (1) ES = Single-end ED = Differential	M - 1 7 1 3 - 1.5 ES 1 0 0
Line count 100, 200, 250, 400, 500 or 1000 (2)	M - 1 7 1 3 - 1.5 ES 1 0 0

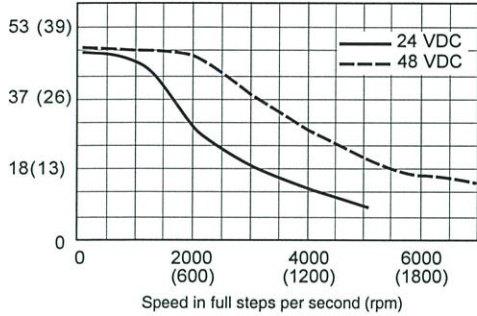
(1) An encoder replaces the shaft designator in the part number.

(2) All encoders have an index mark, except the 1000 line count version.



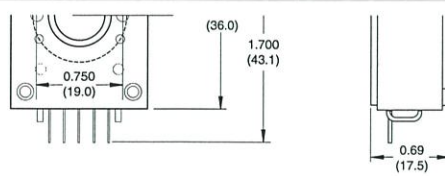
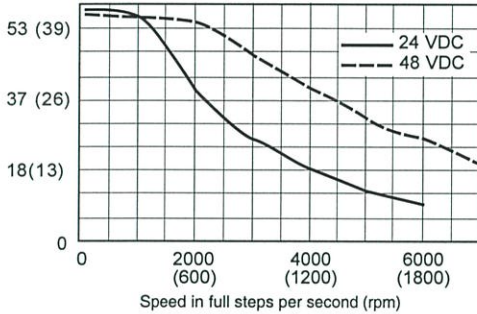
M-1715-1.5

Torque in oz-in (N-cm)

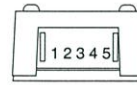


M-1719-1.5

Torque in oz-in (N-cm)



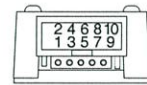
Connectivity single-end encoder



wire	function
1 Brown	Ground
2 Violet	Index
3 Blue	Channel A
4 Orange	+5 VDC input
5 Yellow	Channel B

optional interface cable available: ES-CABLE-2

differential encoder

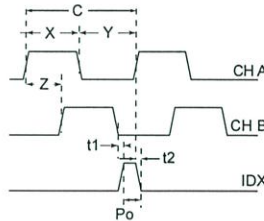


pin function	pin function
1 no connect	6 Channel A+
2 +5 VDC input	7 Channel B-
3 Ground	8 Channel B+
4 no connect	9 Index-
5 Channel A-	10 Index+

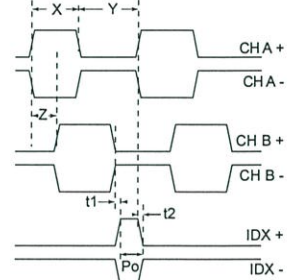
interface cable included

Timing

single-end encoder



differential encoder



Parameter	Symbol	Min	Typ	Max	Units
Cycle error			3	5.5	°e
Symmetry		130	180	230	°e
Quadrature		40	90	140	°e
Index pulse width	Po	60	90	120	°e
Index rise (after Ch A or B rise)	t1	-300	100	250	ns
Index fall (after Ch A or B fall)	t2	70	150	1000	ns

- C One cycle: 360 electrical degrees (°e).
- X/Y Symmetry: the measure of the relationship between X and Y, nominally 180°e.
- Z Quadrature: the phase lead or lag between channels A and B, nominally 90°e.
- Po Index pulse width, nominally 90°e.

NOTE: Rotation is as viewed from the cover side of the encoder.