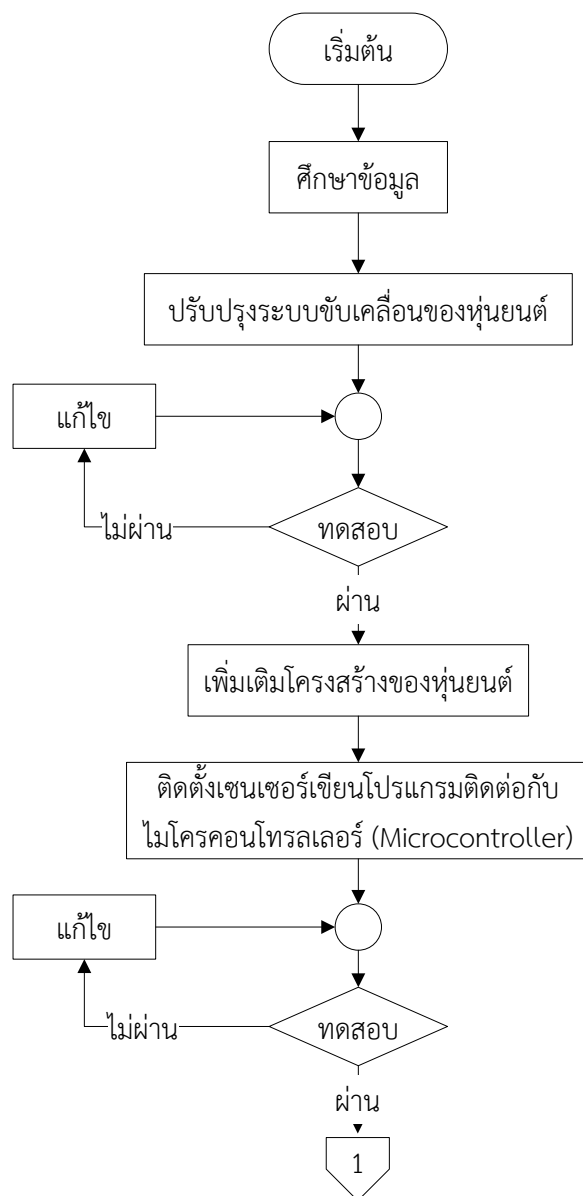


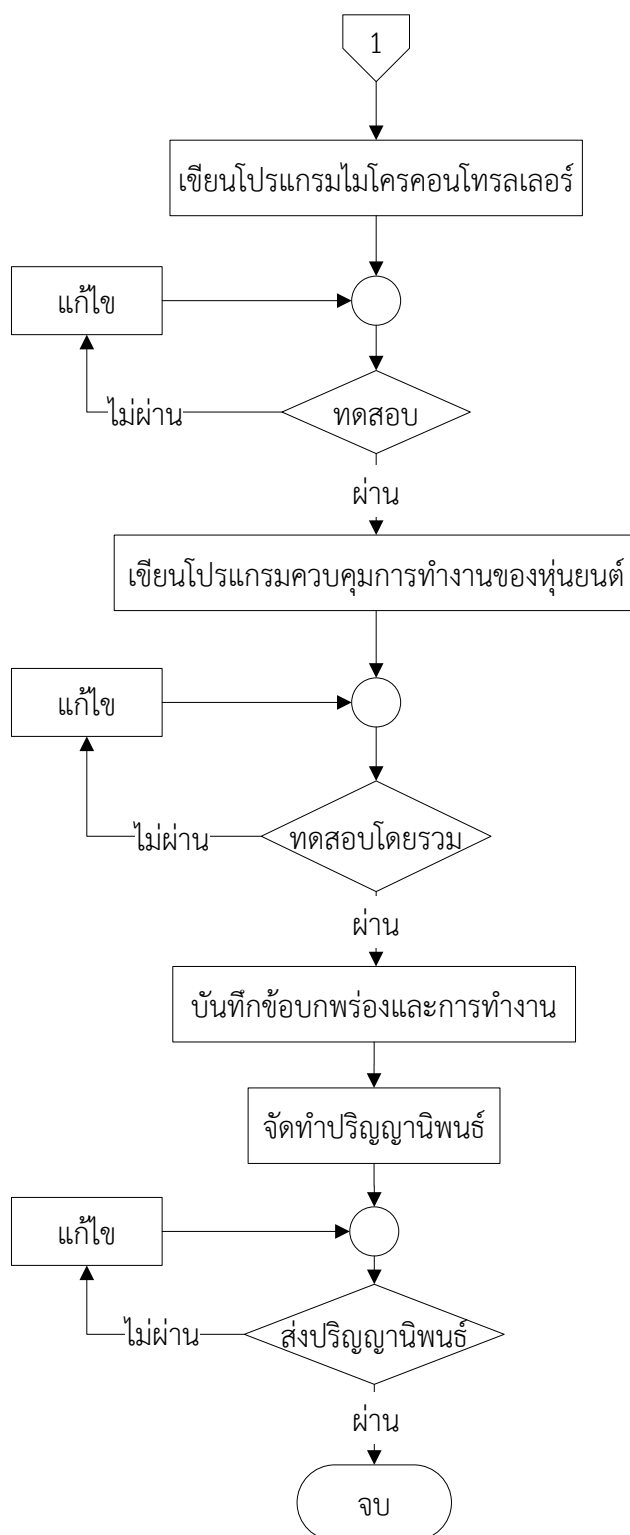
บทที่ 3

ขั้นตอนและวิธีการดำเนินงาน

จากทฤษฎีและข้อมูลที่เกี่ยวข้อง ทางผู้จัดทำได้มีการวางแผนและผังในการปฏิบัติงานดังนี้



ภาพที่ 3-1 ขั้นตอนและวิธีการดำเนินงานของโครงการ



ภาพที่ 3-2 ขั้นตอนและวิธีการดำเนินงานของโครงการ (ต่อ)

ตารางที่ 3-1 ระยะเวลาในการดำเนินงาน

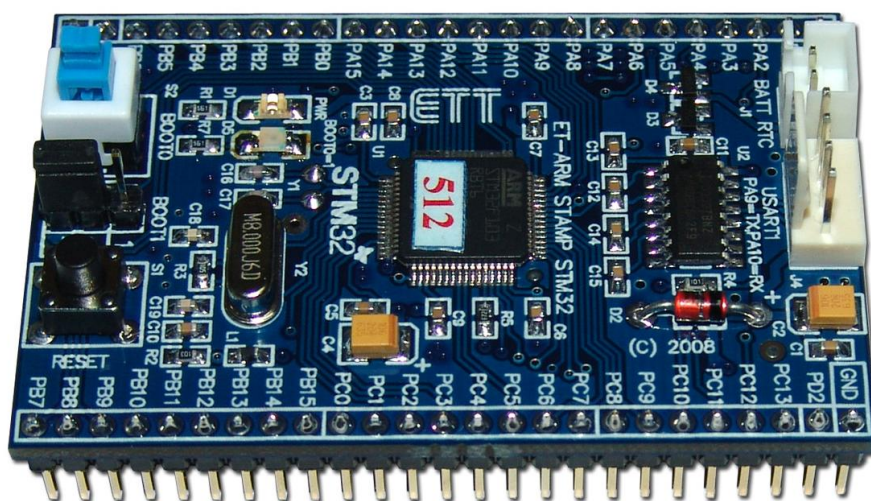
ขั้นตอนการดำเนินงาน	2553												2554					
	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.
ศึกษาข้อมูลเพิ่มเติม																		
ปรับปรุงฮาร์ดแวร์																		
ติดตั้งเซิร์ฟเวอร์																		
เขียนโปรแกรมติดต่อกับ ไมโครคอนโทรลเลอร์																		
เขียนโปรแกรมการ ประมวลผลทางภาพ																		
เขียนโปรแกรมควบคุมการ ทำงานของหุ่นยนต์																		
ทดสอบการทำงานโดยรวม ของหุ่นยนต์ แก้ไข																		
จัดทำปริญญานิพนธ์และ แก้ไข																		
จัดพิมพ์ปริญญานิพนธ์																		

ขั้นตอนในการดำเนินงาน

- 3.1 ศึกษาข้อมูลเพิ่มเติม
- 3.2 ปรับปรุงฮาร์ดแวร์ (Hardware)
- 3.3 ติดตั้งเซนเซอร์
 - 3.3.1 ติดตั้งเข็มทิศอิเล็กทรอนิกส์
 - 3.3.2 ติดตั้งโมดูลวัดความเร่ง
 - 3.3.3 ติดตั้งไจโรสโคป (Gyroscope)
 - 3.3.4 ติดตั้งอัลตราโซนิก (Ultrasonic)
 - 3.3.5 ติดตั้งเอ็นโค้ดเดอร์ (Encoder)
- 3.4 ติดตั้งงานรักษาสมดุลและเขียนโปรแกรมควบคุม
- 3.5 เขียนโปรแกรมสร้างแผนที่ของหุ่นยนต์
- 3.6 เขียนโปรแกรมการประมวลผลทางภาพ
- 3.7 เขียนโปรแกรมควบคุมการทำงานของหุ่นยนต์
- 3.8 ทดสอบการทำงานโดยรวมของหุ่นยนต์

3.1 ศึกษาข้อมูลเพิ่มเติม

- 3.1.1 การใช้งานไมโครคอนโทรลเลอร์ (Microcontroller) รุ่น ET-ARM STAMP STM32



ภาพที่ 3-3 ไมโครคอนโทรลเลอร์ (Microcontroller) รุ่น ET-ARM STAMP STM32

3.1.1.1 คุณสมบัติของบอร์ด รุ่น ET-ARM STAMP STM32

ก) ใช้ไมโครคอนโทรลเลอร์ 32 บิตตระกูล ARM Cortex-M3 ขนาด 64 ขาของ STMicroelectronics

ข) ใช้แหล่งกำเนิดสัญญาณ (Crystal) 8.00 MHz บวก Phase-Locked Loop (PLL) ทำงานความถี่ 72 MHz ประมวลผลด้วยความเร็ว 1.25 DMIPS/MHz เทียบเท่ากับความเร็วการประมวลผลเท่ากับ 90MIPS

ค) มีวงจร RTC (Real Time Clock) พร้อมแหล่งกำเนิดสัญญาณ (XTAL) ค่า 32.768 KHz และแบตเตอรี่สำรอง

ง) รองรับการโปรแกรมแบบ In-System Programming (ISP) และแบบ In-Application-Programming (IAP) ผ่านทางโปรแกรม On-Chip Boot-Loader ทางคอมพอร์ต USART1 (RS232)

จ) มีวงจรสื่อสาร RS232 โดยใช้ขั้วต่อแบบ 4-PIN จำนวน 1 ช่อง

ฉ) มี 48 Bit GPIO พร้อม 5 โวลต์ สำหรับประยุกต์ต่างๆ

ช) ใช้กับแหล่งจ่ายไฟขนาด +3.3 โวลต์

ซ) ขั้วต่อวางตัวบน Pin Header ระยะห่าง 2.54 มิลลิเมตร (100 มิล) ขนาด 50 ขา (ด้านละ 25 ขา) ระยะห่าง 1.5 นิ้ว (1500 มิลต่อ 38.1 มิลลิเมตร) ง่ายต่อการประยุกต์ใช้งาน และขยายวงจร I/O สามารถใช้กับ Project Board และ PCB

3.1.1.2 ตำแหน่งของอุปกรณ์ต่างๆในบอร์ด ET-ARM STAMP STM32

หมายเลข 1 คือ MCU ประจำบอร์ด

หมายเลข 2 คือ ขั้วต่อ USART1 (RS232) สำหรับใช้งาน และ Download

หมายเลข 3 คือ ขั้วต่อ Battery Backup-RTC ขนาด +3 โวลต์

หมายเลข 4 คือ LED Power

หมายเลข 5 คือ LED BOOT0 (BOOT0=1)

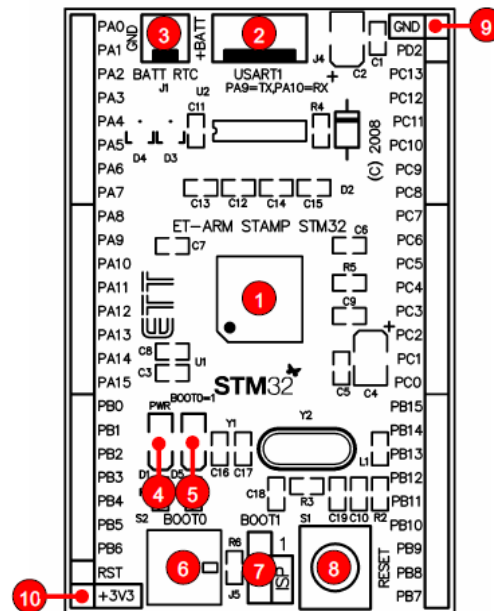
หมายเลข 6 คือ สวิตช์ BOOT0 ซึ่งใช้ร่วมกับ Jumper J1 สำหรับเลือกโหมดการทำงานของบอร์ดแบบ Download จาก USART1 หรือ Run ปกติ โดย ON เพื่อ Download และ OFF เพื่อ Run

หมายเลข 7 คือ Jumper (BOOT1) ใช้ร่วมกับสวิตช์ BOOT0 สำหรับเลือกโหมดการทำงานของบอร์ดแบบ Download จาก USART1 หรือ Run ปกติ ซึ่งต้องกำหนดไว้ที่ ISP เสมอ

หมายเลข 8 คือ สวิตช์ RESET สำหรับ RESET การทำงานของ MCU

หมายเลข 9 คือ จุดต่อ GND

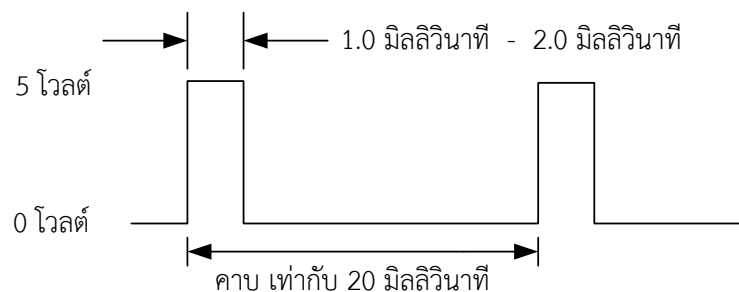
หมายเลข 10 คือ ไฟเลี้ยงวงจร +3.3 โวลต์



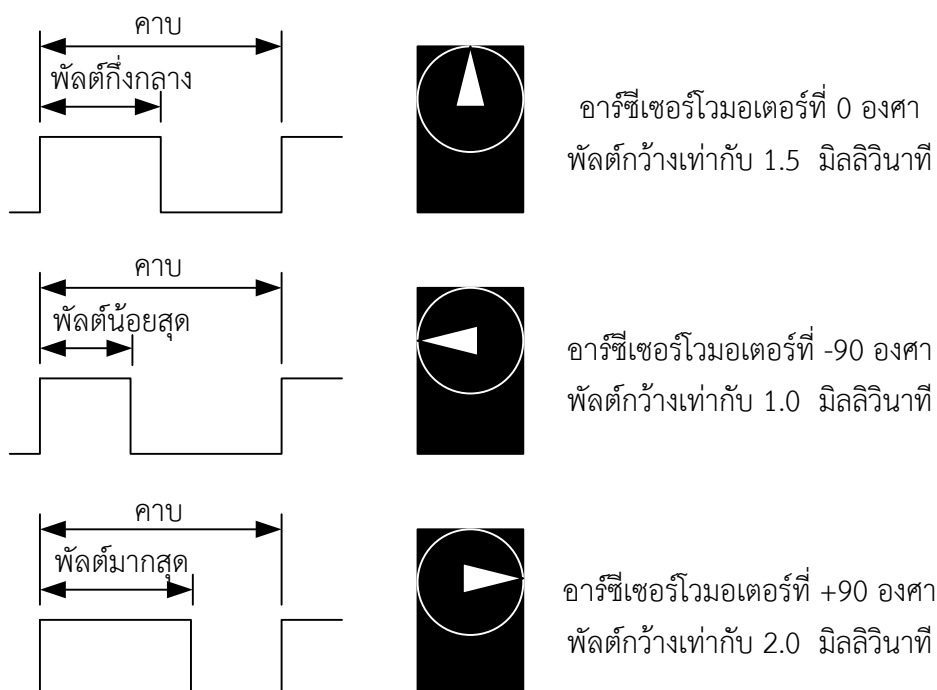
ภาพที่ 3-4 ตำแหน่งของอุปกรณ์ต่างๆในบอร์ด ET-ARM STAMP STM32

3.1.2 การใช้งานอาร์ซีเซอร์โวมอเตอร์ (RC Servo Motor)

การควบคุมการทำงานของอาร์ซีเซอร์โวมอเตอร์ (RC Servo Motor) ทำได้โดยการป้อนสัญญาณความกว้างพัลส์ ให้กับมอเตอร์ซึ่งตำแหน่ง และทิศทางการหมุนของมอเตอร์นี้ จะขึ้นอยู่กับขนาดของความกว้างของพัลส์นั้นๆ โดยทั่วไปแล้วความกว้างของสัญญาณพัลส์จะมีจุดให้อ้างอิง 3 จุด ดังรูป คือ



ภาพที่ 3-5 การป้อนสัญญาณความกว้างพัลส์



ภาพที่ 3-6 จุดอ้างอิงของสัญญาณพัลส์

สัญญาณความกว้างพัลส์ ขนาด 1.5 มิลลิวินาที จะควบคุมให้อาร์ซีเซอร์ไวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม 0 องศา หรือ จุดกึ่งกลางของมอเตอร์

สัญญาณความกว้างพัลส์ ขนาด 1 มิลลิวินาที จะควบคุมให้อาร์ซีเซอร์ไวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม - 90 องศา หรือในทิศทางทวนเข็มนาฬิกา

สัญญาณความกว้างพัลส์ ขนาด 2 มิลลิวินาที จะควบคุมให้อาร์ซีเซอร์ไวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม + 90 องศา หรือในทิศทางตามเข็มนาฬิกา

ส่วนการที่จะควบคุมให้มอเตอร์หมุนเป็นมุมอื่นๆ ก็สามารถทำได้โดยการป้อนสัญญาณพัลส์เป็นระดับความกว้างต่างๆ โดยอ้างอิงจากจุด ทั้ง 3 จุดที่กล่าวมานี้ ตัวอย่าง เช่น ถ้าต้องการให้มอเตอร์หมุนไปที่มุม -45 องศา ต้องป้อนสัญญาณพัลส์ที่มีความกว้าง 1.25 มิลลิวินาที เป็นต้น และสัญญาณพัลส์ ต้องจ่ายให้มอเตอร์ทุกๆ 20 มิลลิวินาที เพื่อรักษาสภาพตำแหน่งของมอเตอร์ให้อยู่ ณ ตำแหน่งไว้

โดยหลักการก็ คือ จะอาศัยการเปรียบเทียบช่วงเวลาของความกว้างพัลส์ที่จ่ายให้กับมอเตอร์ทางขาสัญญาณควบคุมกับค่าเวลาของวงจรอาร์ซี (RC) ภายในบอร์ดควบคุมในตัวของมอเตอร์ ซึ่งค่าเวลาของวงจรอาร์ซี นี้จะมีการเปลี่ยนแปลงตามการหมุนของมอเตอร์ เนื่องจากตัวต้านทานปรับค่า

จะถูกยึดติดอยู่กับแกนหมุนของมอเตอร์ ซึ่งการหมุนของมอเตอร์จะทำให้ค่าความต้านทานของตัวต้านทานปรับค่า (Variable Resistors : VR) เปลี่ยนแปลงไป เป็นผลทำให้ค่าเวลาของวงจรอาร์ซีเปลี่ยนแปลงตามไปด้วย โดยในขณะที่เราป้อนสัญญาณความกว้างพัลส์ให้กับมอเตอร์ทางขาสัญญาณควบคุม สัญญาณนี้จะถูกนำไปเปรียบเทียบกับค่าเวลาของวงจรอาร์ซี หากค่าทั้ง 2 ไม่เท่ากันมอเตอร์ก็จะหมุนทำให้ค่าเวลาของวงจรอาร์ซี เปลี่ยนแปลงจนกระทั่งค่าเวลาความกว้างพัลส์ของ วงจรอาร์ซีเปลี่ยนแปลงจนเท่ากับสัญญาณพัลส์ทางขาควบคุม (Control Line) มอเตอร์จึงจะหยุดหมุน

3.2 ปรับปรุงฮาร์ดแวร์ (Hardware)

3.2.1 ชุดเกียร์ (Gear Box) เริ่มต้นได้นำหุ่นยนต์ที่ทางสาขาวิชาแมคคาทรอนิกส์ที่มีอยู่แล้วมาทำการพัฒนาต่อ และพบปัญหาที่ชุดเกียร์ (Gear Box) ประการแรกคือเมื่อเกิดแรงบิดสูงๆแกนเพลลาของ ชุดเกียร์ (Gear Box) จะเสีयरูปทันที สิ่งที่มาคือต้องเสียเวลาและงบประมาณในการซ่อมแซม ประการที่สองชุดเกียร์ (Gear Box) ที่มีอยู่ ได้ผ่านการใช้งานมาจนมีสภาพที่ไม่สามารถทำงานได้อย่างมีประสิทธิภาพ จึงได้ปรับเปลี่ยนระบบขับเคลื่อนจาก ชุดเกียร์ (Gear Box) มาเป็นระบบการขับเคลื่อนด้วยเฟืองเกียร์ โดยใช้อัตราทด 20:75 ฟัน



ภาพที่ 3-7(ก) ชุดเกียร์ที่ติดตั้งอยู่ก่อนหน้า



ภาพที่ 3-7(ข) ชุดเฟืองเกียร์ที่ติดตั้งใหม่

3.2.2 ชุดล้อ หลังจากที่ได้ปรึกษากับอาจารย์ที่ปรึกษาโครงการ อาจารย์ได้ให้คำแนะนำว่าควรที่จะเปลี่ยนล้อ คือเปลี่ยนจากเดิมเป็นแบบสายพานให้ใช้เป็นแบบล้อกลม เนื่องจากลดความยุ่งยากในการประกอบ เคลื่อนที่ได้คล่องตัว บำรุงรักษาได้ง่ายกว่า มีราคาถูก ลดปัญหาเรื่องสายพานหย่อน และในส่วนการทำงานของหุ่นยนต์อัตโนมัติ (Automatic) ไม่ได้มีการป็นปายในส่วนพื้นที่ที่ยากลำบาก จึงไม่จำเป็นต้องมีโครงสร้างที่ซับซ้อน

ตารางที่ 3-2 การให้คะแนนของระบบขับเคลื่อนแบบล้อและแบบสายพาน

หัวข้อ	ระบบล้อ	ระบบสายพาน
พื้นผิวสัมผัส	7	10
การประกอบ	10	7
การส่งกำลัง	8	9
ราคา	10	5
การบำรุงรักษา	9	7
ความแข็งแรง	7	9
อายุการใช้งาน	8	8

3.2.3 มอเตอร์ (Motor) เนื่องจากมอเตอร์ที่มีอยู่ไม่สามารถใช้งานได้เพราะมีกำลังขับไม่เพียงพอ จึงต้องหามอเตอร์ขนาดที่มีกำลังมากกว่ามอเตอร์เดิมมาใช้กับหุ่นยนต์ โดยหลักการเลือกใช้งานมอเตอร์คือ

$$\begin{aligned}
 &\text{ค่าความปลอดภัย Safety factor (S.F.)} &= & 2.5 \\
 &\text{หุ่นยนต์มีน้ำหนักรวม} &= & 20 \times 9.81 \times 2.5 \\
 & &= & 490.5 && \text{นิวตัน} \\
 &\text{อัตราทดของเฟือง} &= & 1:3.75 && \text{เท่า} \\
 &\text{ความเร็วรอบ} &= & \frac{500}{3.75} \\
 & &= & 133.33 && \text{รอบต่อนาที} \\
 &\text{เส้นผ่านศูนย์กลางล้อ} &= & 150 && \text{มิลลิเมตร} \\
 &\text{แปลงเป็นหน่วยเมตร} &= & 0.15 && \text{เมตร} \\
 &\text{การหมุน 1 รอบของล้อได้ระยะทาง} &= & \pi \times 0.15 \\
 & &= & 0.471 && \text{เมตร} \\
 &\text{แรงบิด (Torque) ของล้อที่ใช้} &= & \frac{490.5 \times 0.15}{3.75} \\
 & &= & 19.62 && \text{นิวตันเมตร}
 \end{aligned}$$



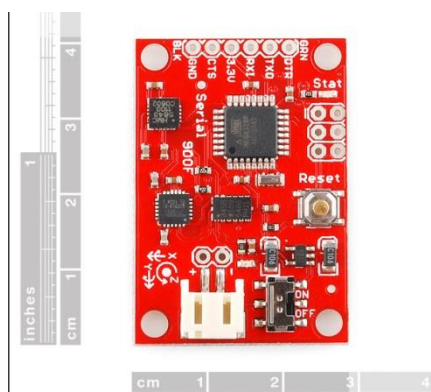
ภาพที่ 3-8 มอเตอร์กระแสตรงที่ใช้

3.3 ติดตั้งเซ็นเซอร์เขียนโปรแกรมติดกับไมโครคอนโทรลเลอร์ (Microcontroller)

3.3.1 ติดตั้งเซ็นเซอร์อิเล็กทรอนิกส์ในการเคลื่อนที่ของหุ่นยนต์ ต้องอาศัยการทำงานของเซ็นเซอร์อิเล็กทรอนิกส์เพื่อที่จะนำตำแหน่งที่ได้มากำหนดทิศทางการเคลื่อนที่ของหุ่นยนต์ โดยได้เลือกใช้ติดตั้งเซ็นเซอร์อิเล็กทรอนิกส์รุ่น HMC6352

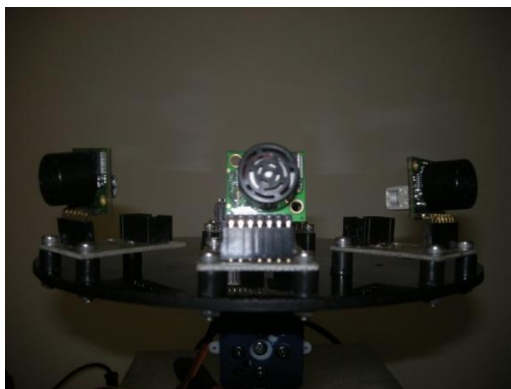
3.3.2 โมดูลวัดความเร็ว เป็นเซ็นเซอร์วัดความเร็ว 3 แกน เซ็นเซอร์วัดความเร็วเชิงมุม 3 แกน แล้วส่งค่าความเร็ว และความเร็วเชิงมุมในการเคลื่อนที่ของทั้ง 3 แกนออกมา โดยจะมีการกรองสัญญาณและการชดเชยค่าผิดพลาดจากการเลื่อนของค่าศูนย์ (Drift)

3.3.3 ไจโรสโคป (Gyroscope) การใช้ฮัลต์ราโซนิกสร้างแผนที่ระบุตำแหน่งโดยที่จะต้องให้ฮัลต์ราโซนิกตั้งขนานกับพื้นผิวอยู่ตลอดเวลา โดยต้องมีการนำฮัลต์ราโซนิกติดตั้งบนจานรักษาสมดุลและใช้ไจโรสโคปเป็นเซ็นเซอร์ในการวัดความเอียงของตัวหุ่นยนต์ในขณะวิ่ง



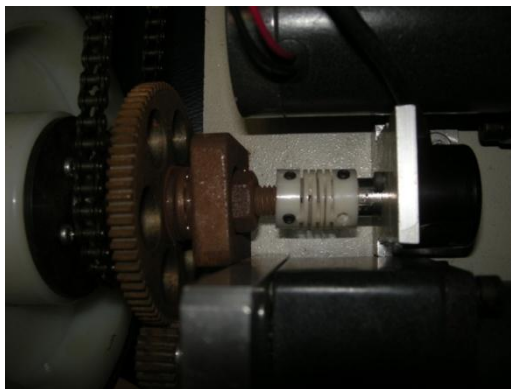
ภาพที่ 3-9 ลักษณะการติดตั้งติดตั้งเซ็นเซอร์อิเล็กทรอนิกส์ โมดูลวัดความเร็ว และไจโรสโคป

3.3.4 อัลตราโซนิก (Ultrasonic) ในการสร้างแผนที่ระบุตำแหน่งของหุ่นยนต์จะใช้อัลตราโซนิก ในการสร้างแผนที่ โดยนำอัลตราโซนิกมา 3 ตัวมาติดตั้งบนจานรักษาสมดุล เพื่อให้แผนที่ที่ได้มีความละเอียดมากยิ่งขึ้น โดยอัลตราโซนิกที่นำมาใช้สามารถวัดระยะได้ตั้งแต่ 0.2 ถึง 10.24 เมตร



ภาพที่ 3-10 การติดตั้งอัลตราโซนิกบนจานรักษาสมดุล

3.3.5 เอ็นโค้ดเดอร์ (Encoder) ขณะที่หุ่นยนต์วิ่งกำลังวิ่งนั้นระยะทางจะมีการเปลี่ยนแปลง อยู่ตลอดเวลา ดังนั้นจึงต้องนำเอ็นโค้ดเดอร์มาใช้วัดระยะการเคลื่อนที่ของหุ่นยนต์ โดยติดตั้งอยู่กับ เพลาด้านหลังของล้อ ในที่นี้ใช้เอ็นโค้ดเดอร์ 400 พัลส์ (Pulse)

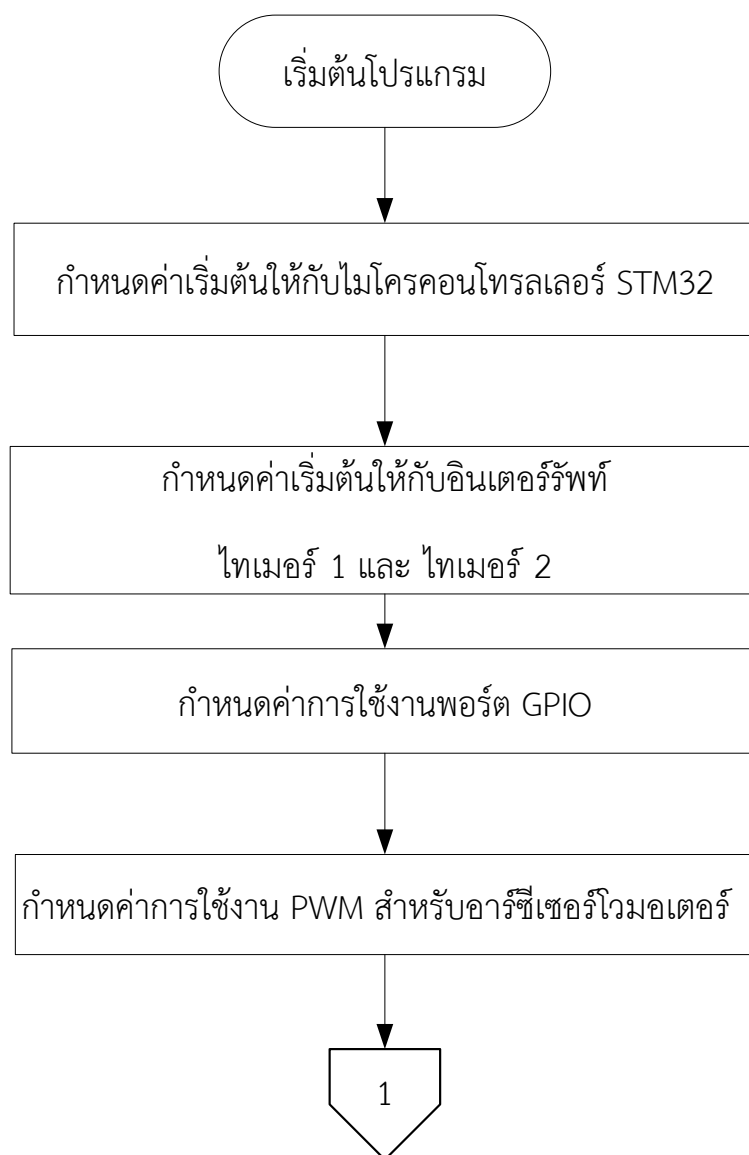


ภาพที่ 3-11 การติดตั้งเอ็นโค้ดเดอร์กับเพลาล้อ

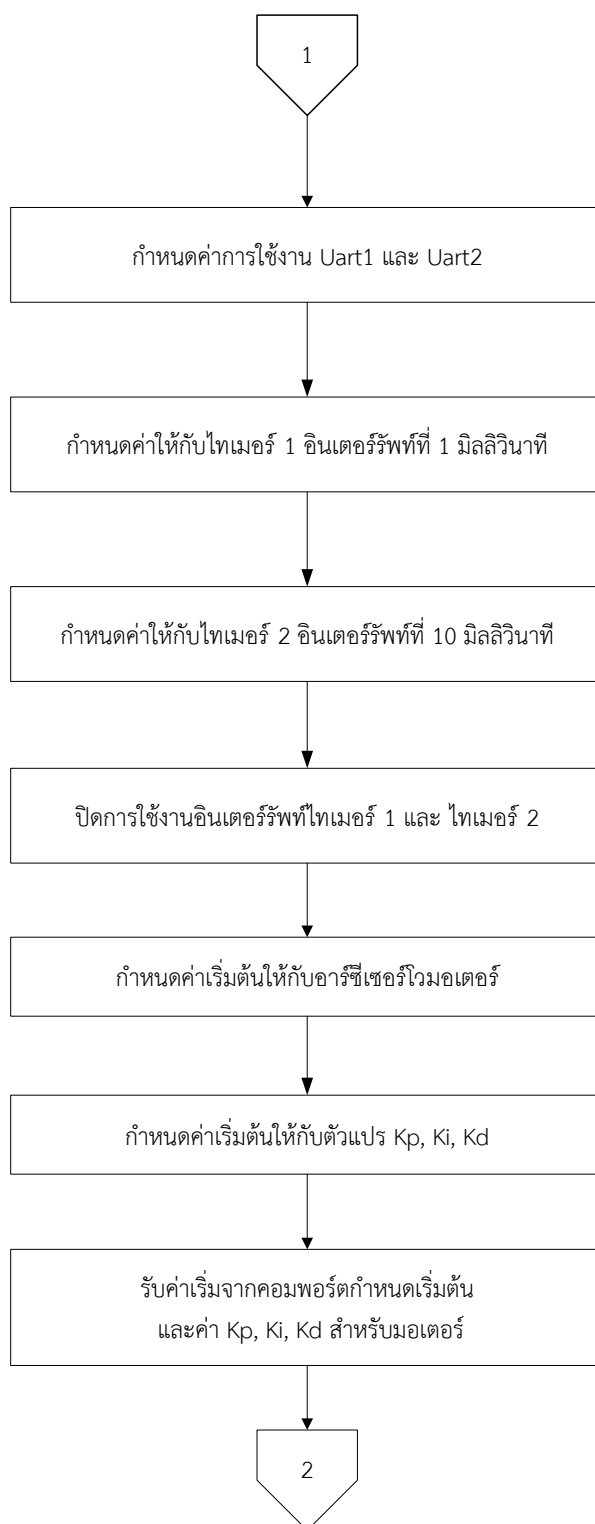
3.4 ติดตั้งจานรักษาสมดุลและเขียนโปรแกรมควบคุม

การทำงานของระบบรักษาสมดุลมีการใช้ไจโรสโคป เป็นเซนเซอร์ที่ใช้ในการวัดความเอียงและ ใช้อาร์ชีเซอร์โมเตอร์ จำนวน 2 ตัว ในการปรับสมดุลของจาน โดยการเขียนโปรแกรม ใช้ไมโครคอนโทรลเลอร์ ET-ARM STAMP STM32 จำนวน 1 ตัว ควบคุมจาน เพื่อทำงานแยกเป็นอิสระต่อกันและง่ายในการเขียนโปรแกรมควบคุม การใช้งานโดยสามารถอธิบายการทำงานได้ดังนี้

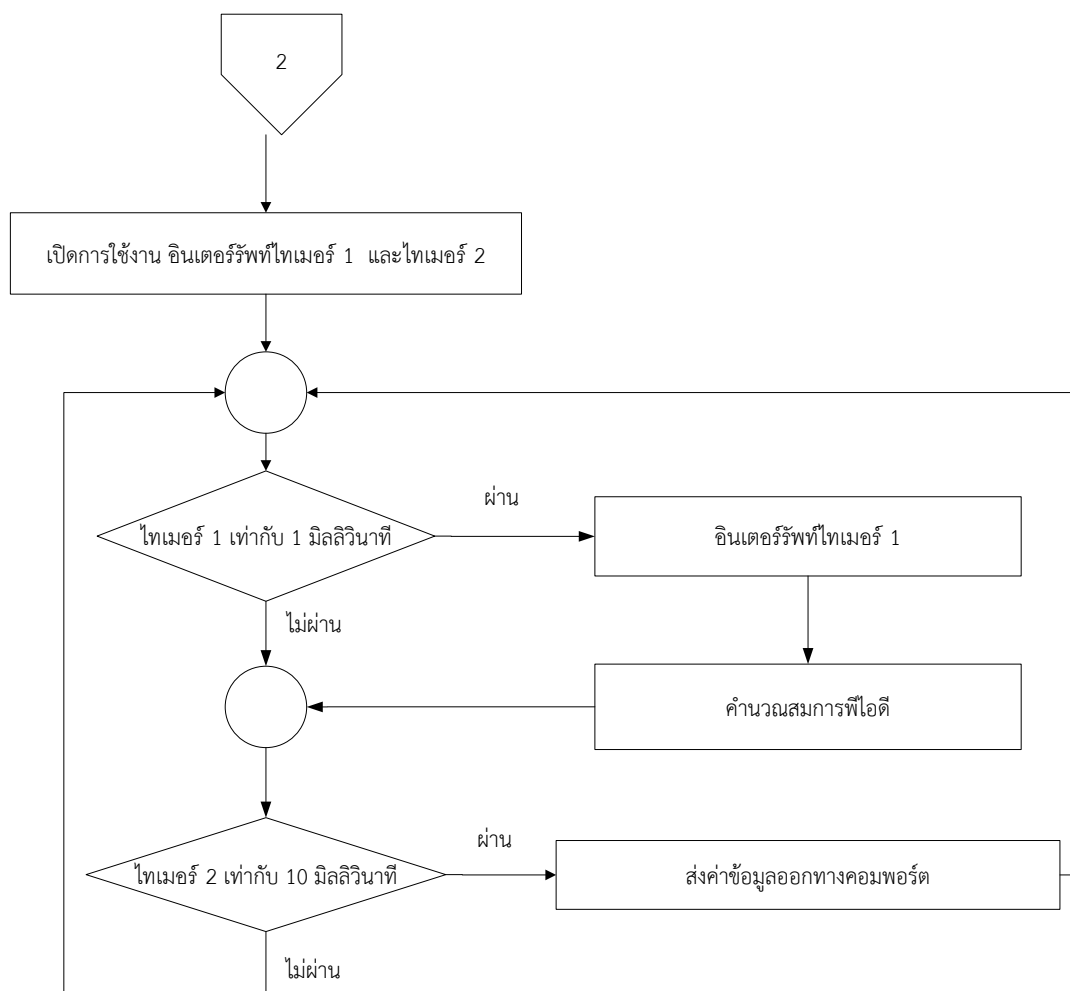
3.4.1 อธิบายการทำงานไมโครคอนโทรลเลอร์ส่วนของระบบงานรักษาสมดุล



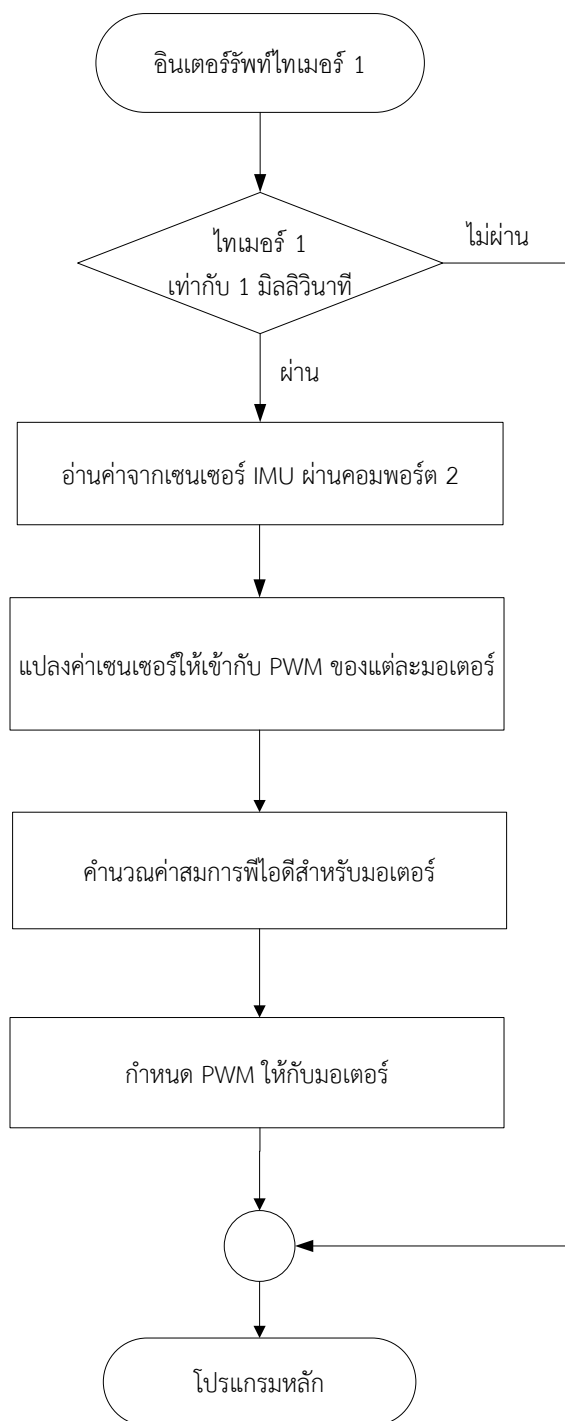
ภาพที่ 3-12 ขั้นตอนการทำงานไมโครคอนโทรลเลอร์ส่วนของระบบงานรักษาสมดุล



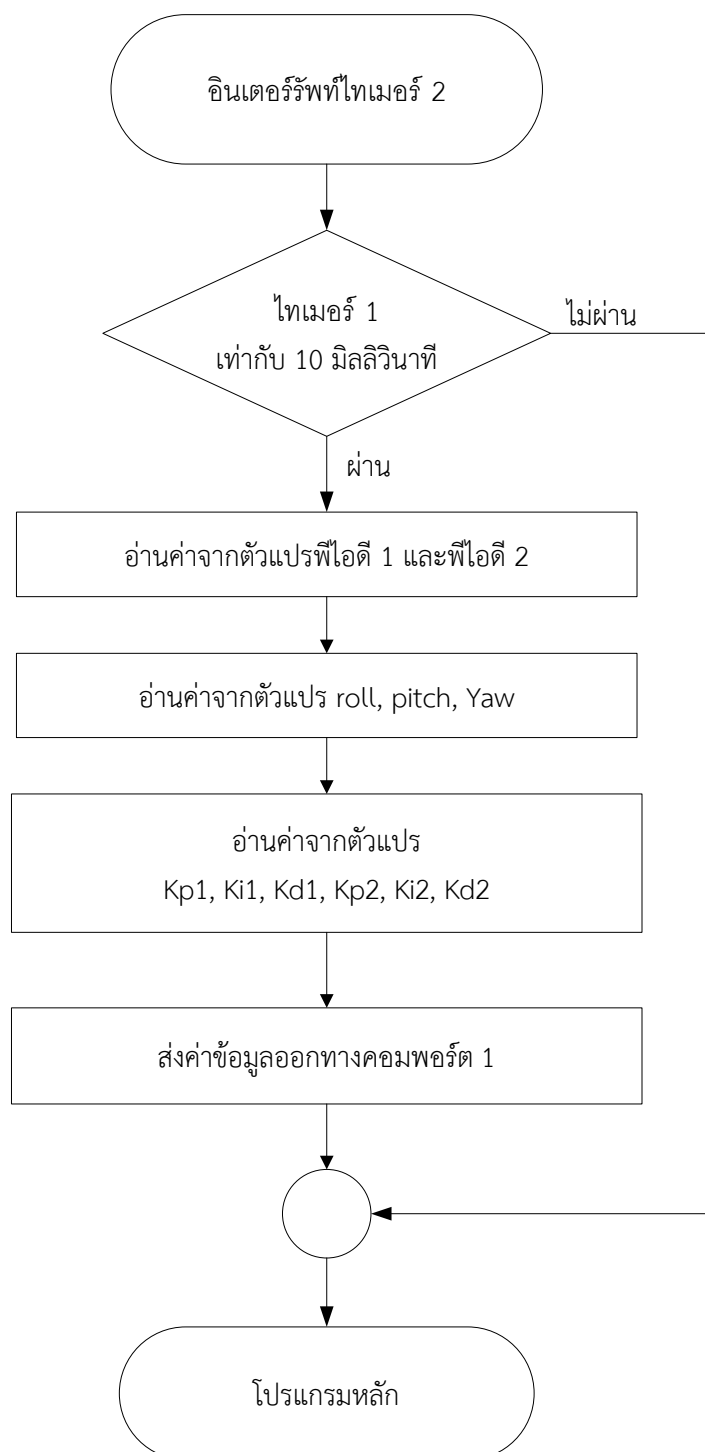
ภาพที่ 3-13 ขั้นตอนการทำงานไมโครคอนโทรลเลอร์ส่วนของระบบงานรักษาสมดุล (ต่อ)



ภาพที่ 3-14 ขั้นตอนการทำงานไมโครคอนโทรลเลอร์ส่วนของระบบงานรักษาสมดุล (ต่อ)

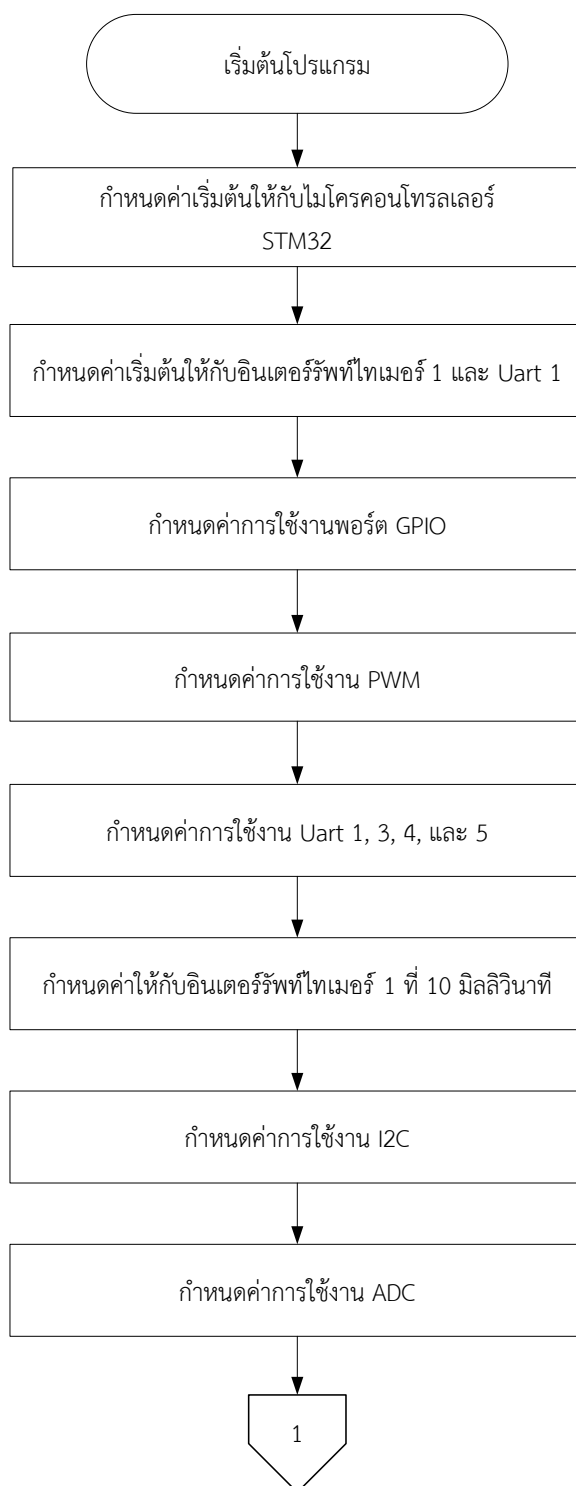


ภาพที่ 3-15 ขั้นตอนการทำงานไมโครคอนโทรลเลอร์ส่วนของระบบงานรักษาสมดุล (ต่อ)

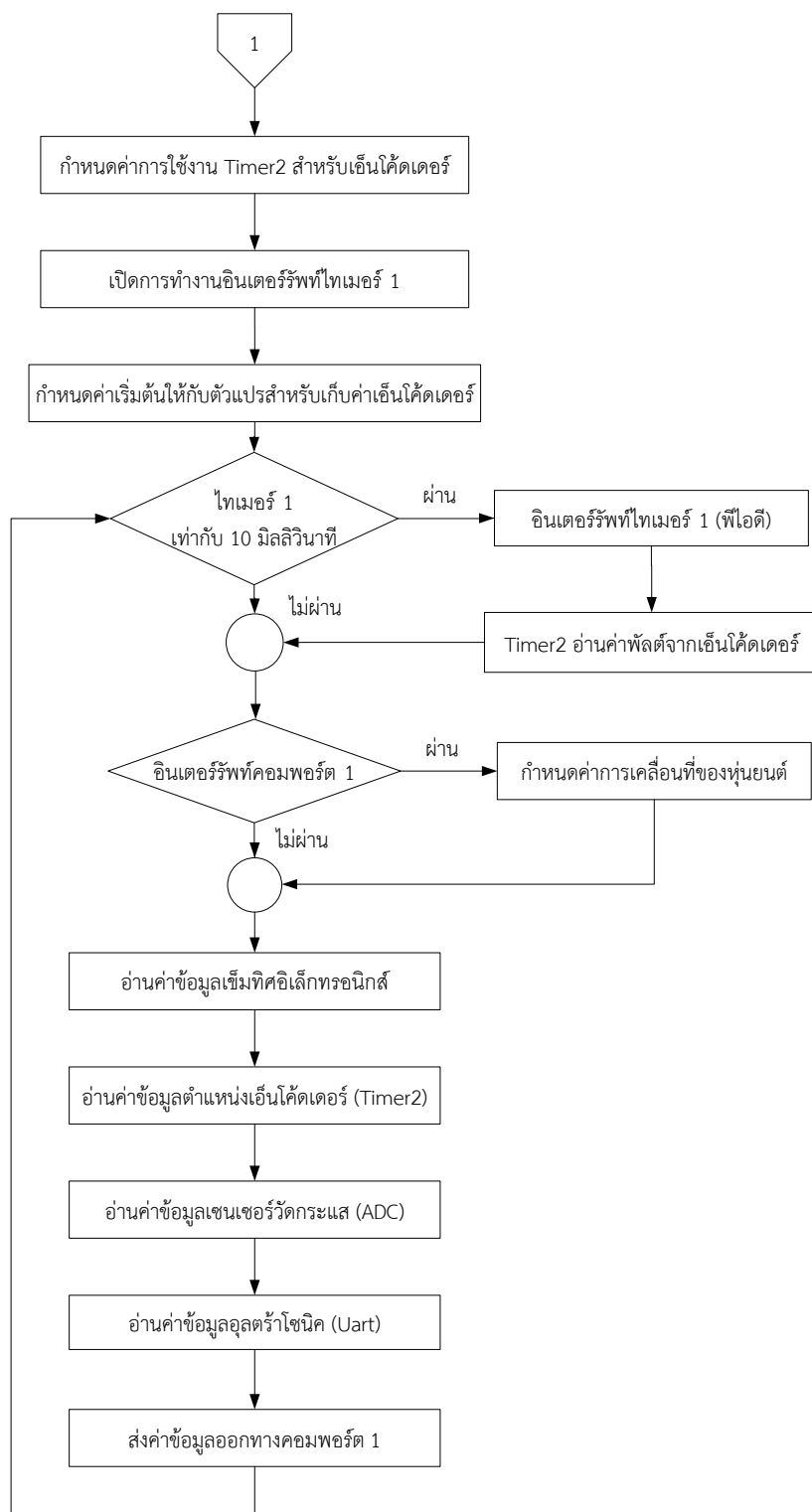


ภาพที่ 3-16 ขั้นตอนการทำงานไมโครคอนโทรลเลอร์ส่วนของระบบจากรักษาสมดุล (ต่อ)

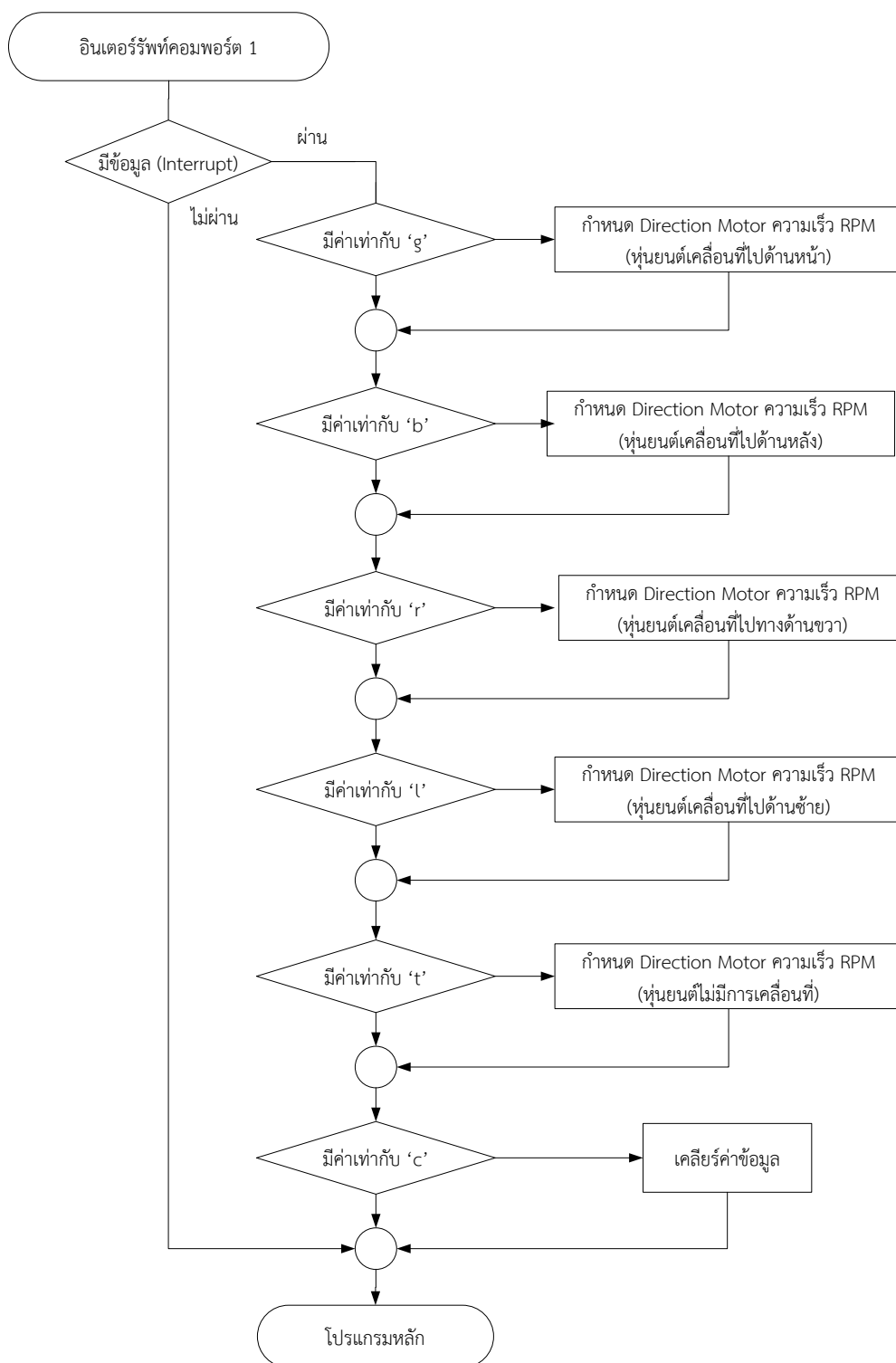
3.4.2 อธิบายการทำงานของไมโครคอนโทรลเลอร์ส่วนของการควบคุมการเคลื่อนที่ของหุ่นยนต์



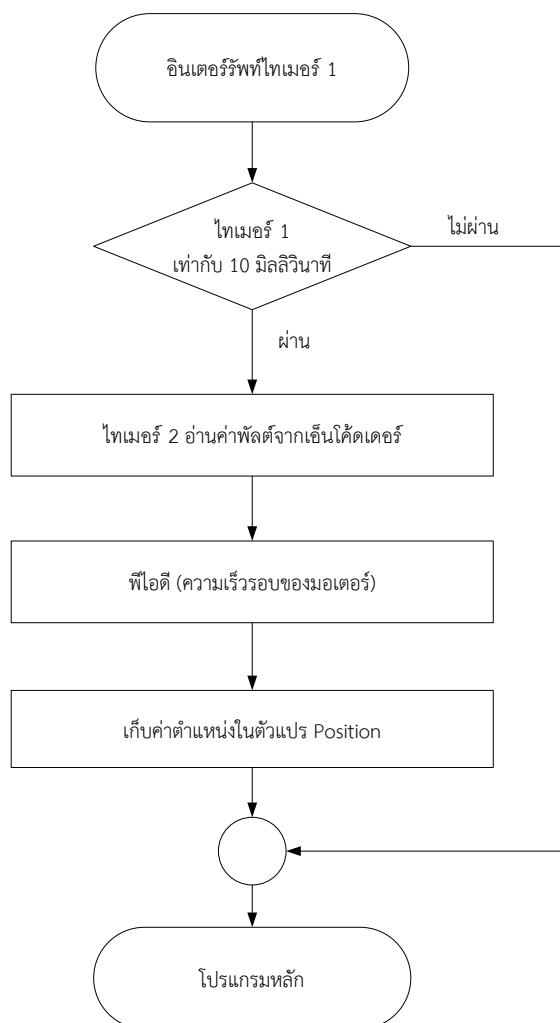
ภาพที่ 3-17 ขั้นตอนการทำงานของไมโครคอนโทรลเลอร์ส่วนของระบบขับเคลื่อนหุ่นยนต์



ภาพที่ 3-18 ขั้นตอนการทำงานไมโครคอนโทรลเลอร์ส่วนของระบบขับเคลื่อนหุ่นยนต์ (ต่อ)



ภาพที่ 3-19 ขั้นตอนการทำงานไมโครคอนโทรลเลอร์ส่วนของระบบขับเคลื่อนหุ่นยนต์ (ต่อ)

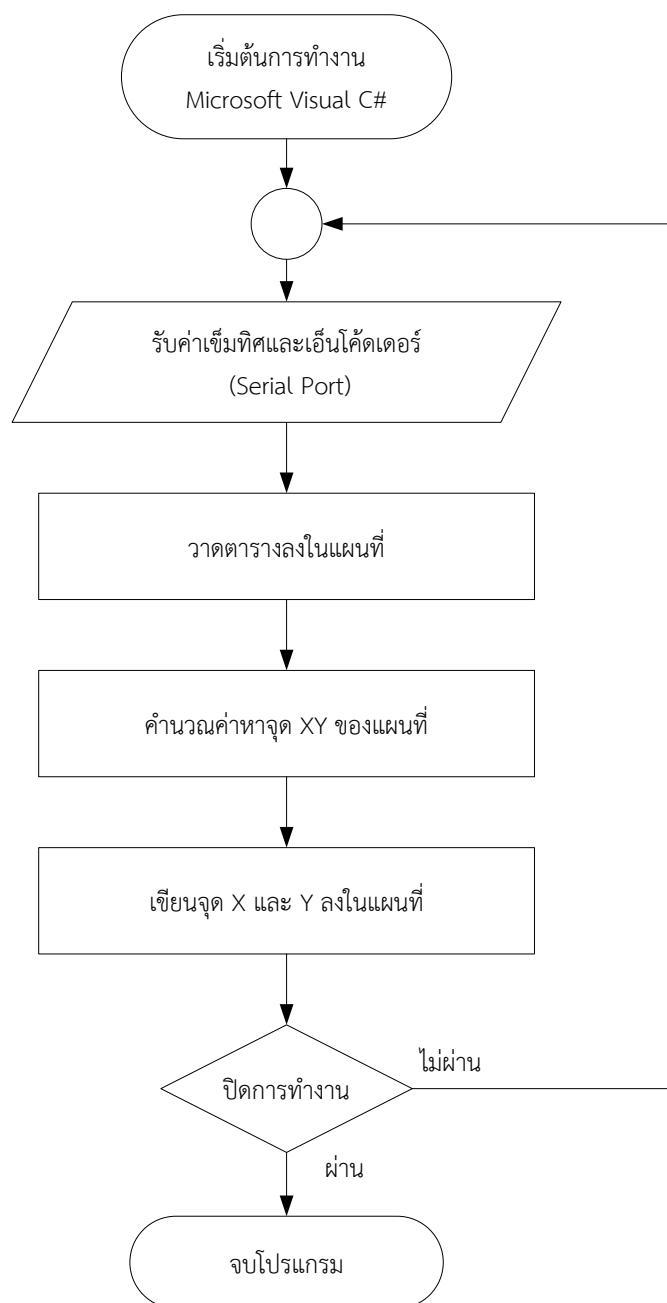


ภาพที่ 3-20 ขั้นตอนการทำงานไมโครคอนโทรลเลอร์ส่วนของระบบขับเคลื่อนหุ่นยนต์ (ต่อ)

3.5 เขียนโปรแกรมสร้างแผนที่ของหุ่นยนต์

การเขียนโปรแกรมในส่วนนี้ ใช้โปรแกรมไมโครซอฟท์วิซวลซีชาร์ป (Microsoft Visual C#) ซึ่งใช้ภาษาซีชาร์ป (C#) ในการเขียนในส่วนของซอร์สโค้ด (Source Code) และการออกแบบส่วนต่อประสานกราฟิกกับผู้ใช้ (GUI) ทำงานร่วมกับไลบรารี GDI+ ที่เป็นส่วนหนึ่งของ .NET Framework ไลบรารีดังกล่าวสามารถสร้างกราฟิกได้อย่างสมบูรณ์ ทั้งเส้นตรง เส้นโค้ง รูปภาพ การระบายสี แบบต่างๆในเบื้องต้น จึงเหมาะสมในการนำมาใช้งานในส่วนของการวาดแผนที่ของหุ่นยนต์ สามารถอธิบายการทำงานได้ดังนี้

3.5.1 อธิบายการทำงานการสร้างแผนที่ของหุ่นยนต์

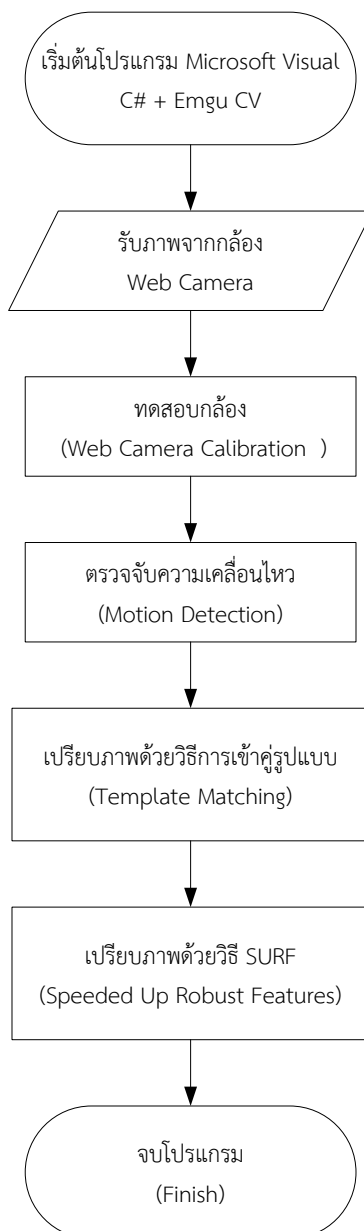


ภาพที่ 3-21 ขั้นตอนการทำงานโปรแกรมสร้างแผนที่ของหุ่นยนต์

3.6 เขียนโปรแกรมการประมวลผลทางภาพ

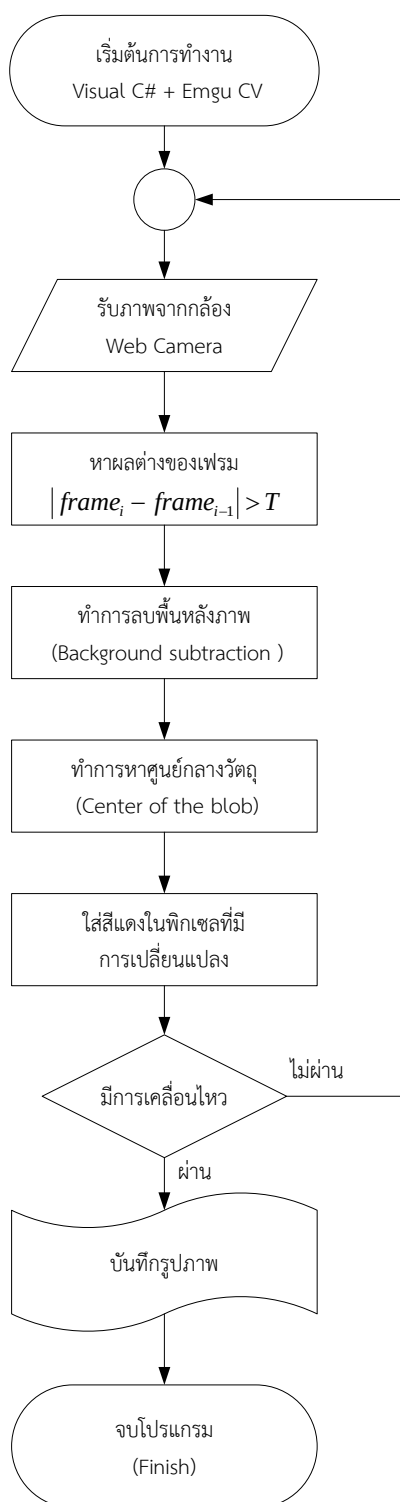
การเขียนโปรแกรมในส่วนนี้ใช้ไลบรารี Emgu CV ซึ่งสามารถทำงานได้หลายระบบปฏิบัติการ ไม่ว่าจะเป็น Windows, Linux, Mac OS X, iPhone, iPad และ Android ซึ่งโปรแกรมนี้ใช้ระบบปฏิบัติการ Windows และใช้ภาษา C# ในการเขียน

3.6.1 อธิบายการทำงานการประมวลผลทางภาพของหุ่นยนต์



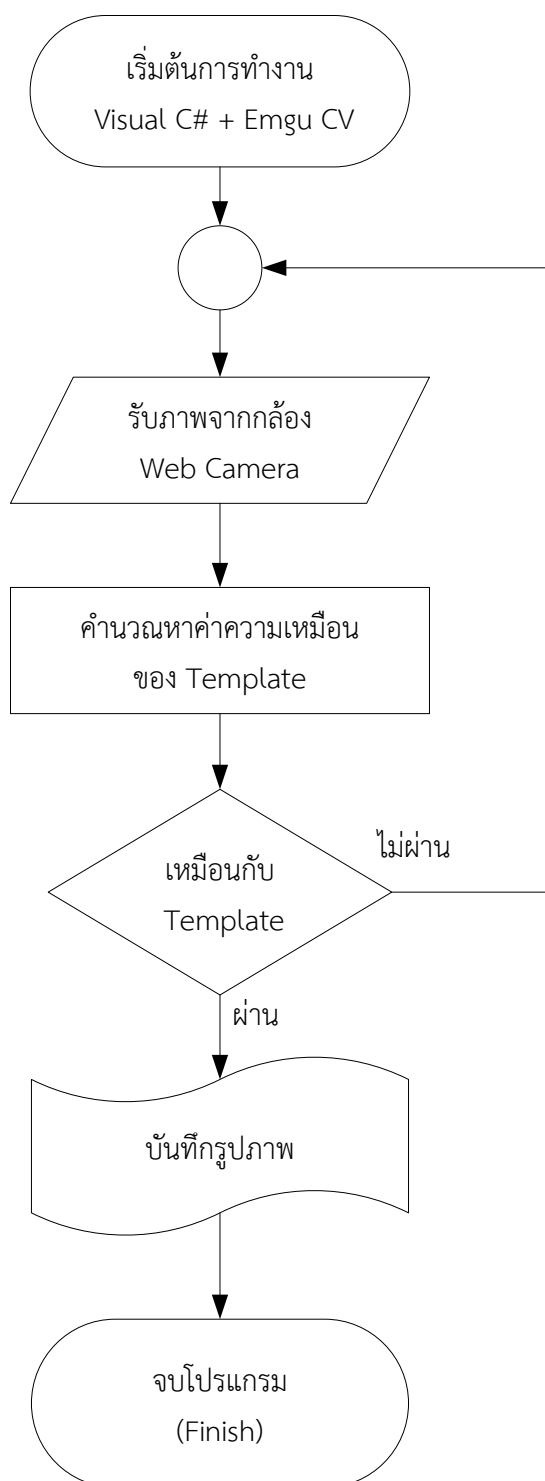
ภาพที่ 3-22 ขั้นตอนการทำงานโปรแกรมการประมวลผลทางภาพ

3.6.2 อธิบายการทำงานวิธีตรวจจับความเคลื่อนไหว (Motion Detection)



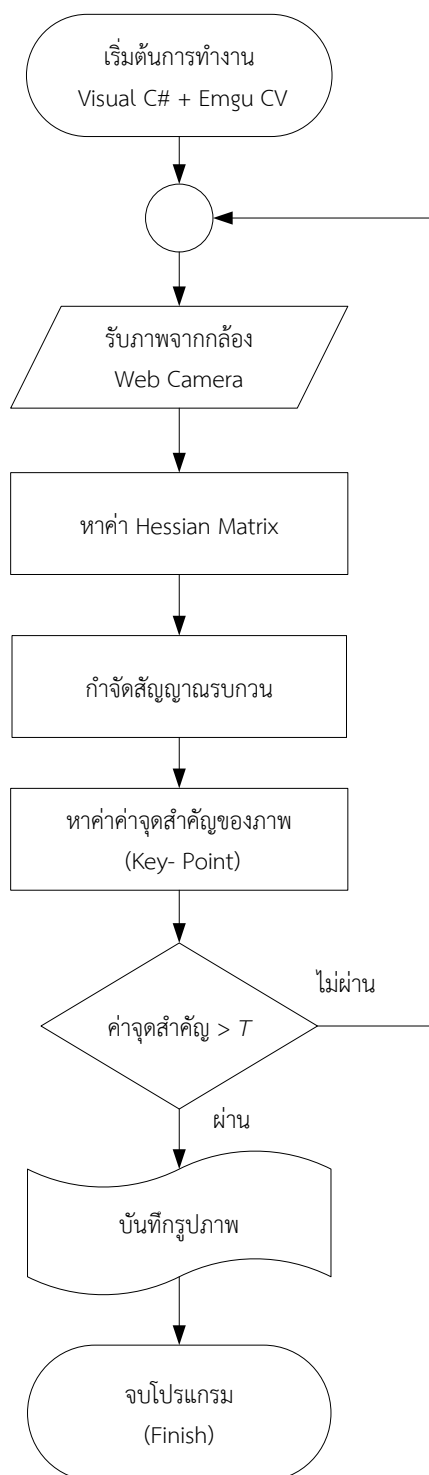
ภาพที่ 3-23 ขั้นตอนการทำงานวิธีตรวจจับความเคลื่อนไหว

3.6.3 อธิบายการทำงานวิธีการเข้าคู่รูปแบบ (Template Matching)



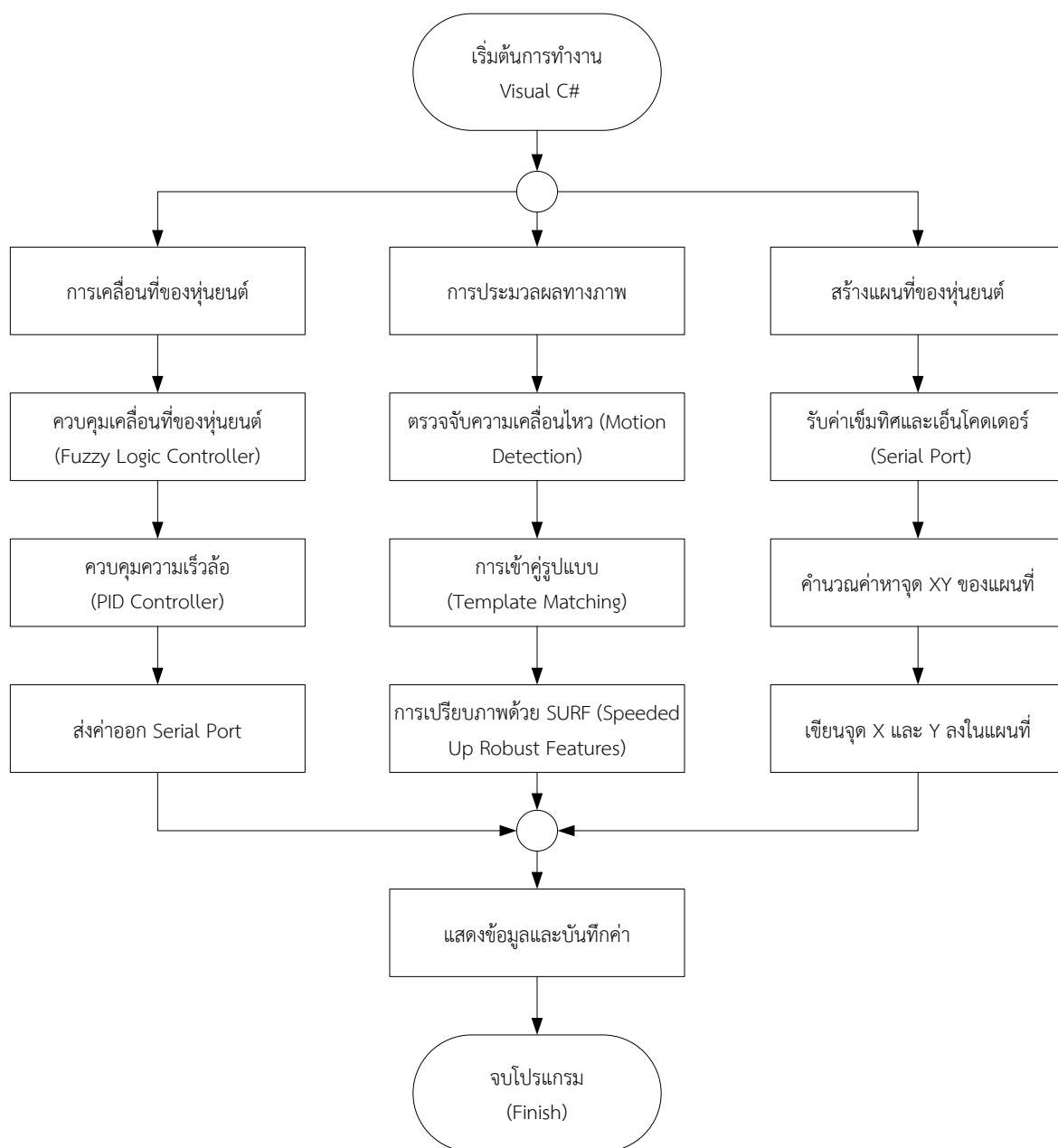
ภาพที่ 3-24 ขั้นตอนการทำงานวิธีการเข้าคู่รูปแบบ

3.6.4 อธิบายการเปรียบเทียบภาพด้วยวิธี SURF (Speeded Up Robust Features)



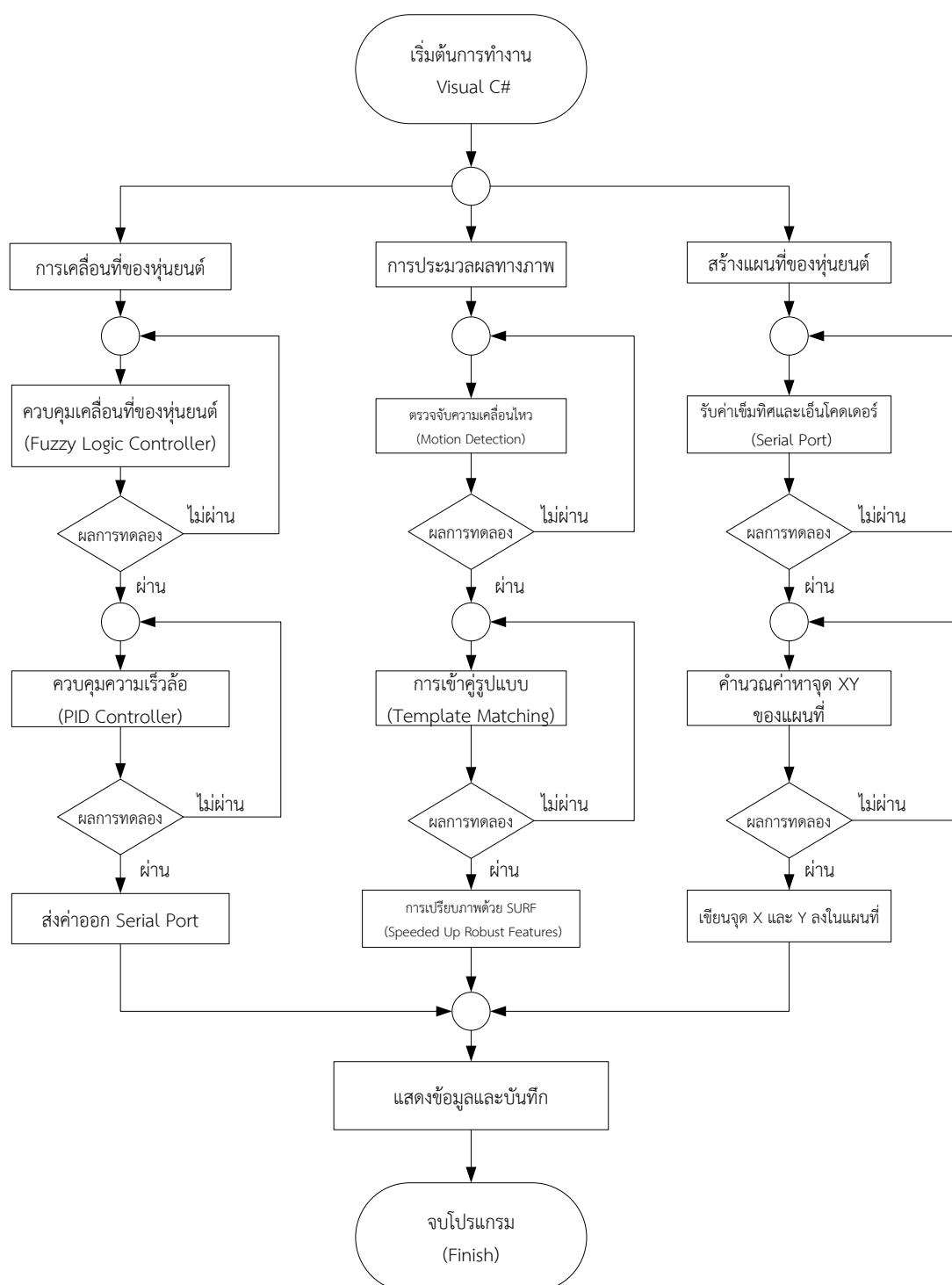
ภาพที่ 3-25 ขั้นตอนการเปรียบเทียบภาพด้วยวิธี SURF

3.7 เขียนโปรแกรมควบคุมการทำงานของหุ่นยนต์



ภาพที่ 3-26 ขั้นตอนการเขียนโปรแกรมควบคุมการทำงานของหุ่นยนต์

3.8 ทดสอบการทำงานโดยรวมของหุ่นยนต์



ภาพที่ 3-27 ขั้นตอนการทดสอบการทำงานโดยรวมของหุ่นยนต์