



รายงานสหกิจศึกษาฉบับสมบูรณ์

ระบบควบคุมการวิ่งสำหรับรถ AGV
TRAFFIC CONTROL SYSTEM FOR AGV

ศตวรรษ รุ่งอรุณวงศ์

ภาควิชาวิศวกรรมการวัดและควบคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2562

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงการ	ระบบควบคุมการวิ่งสำหรับรถ AGV
นักศึกษา	นายศตวรรษ รุ่งอนวงศ์
ภาควิชา	วิศวกรรมการวัดและควบคุม
อาจารย์นิเทศ	ผู้ช่วยศาสตราจารย์ ดร.นพดล มณีรัตน์
ผู้นิเทศงาน	นายอภิวัฒน์ เรืองดิษฐ์
สถานประกอบการ	บริษัท เอสซีจี แพคเกจจิ้ง จำกัด (มหาชน)

บทคัดย่อ

บริษัท เอสซีจี แพคเกจจิ้ง จำกัด (มหาชน) ดำเนินการธุรกิจในการนำเสนอ สินค้าและบริการด้านบรรจุภัณฑ์ โครงการที่ได้รับมอบหมายคือ ระบบควบคุมการวิ่งสำหรับรถ AGV ซึ่งบริษัทมีแนวคิดที่จะลดจำนวนพนักงาน โดยการนำ Automated Guided Vehicle (AGV) มาใช้ในสายการผลิตเพื่อทดแทนพนักงานในการเคลื่อนย้ายสินค้า และเมื่อทำงานมาได้ระยะหนึ่งพบว่า AGV เพียง 1 คัน ไม่เพียงพอต่อการเคลื่อนย้ายสินค้า ซึ่งจะทำให้ใช้เวลาในการเคลื่อนย้ายสินค้าเพิ่มขึ้นหากไม่ทำการแก้ไขปัญหานี้ทางบริษัทจึงมีความต้องการที่จะเพิ่มจำนวน AGV มากกว่า 1 คัน โดยได้เล็งเห็นถึงปัญหาที่จะตามมาหากไม่มีระบบควบคุมการทำงานของ AGV จะทำให้เกิดการชนกันเองหรือชนพนักงานในสายการผลิตได้ จึงมีแนวคิดเกิดเป็นโครงการนี้ขึ้นมา ซึ่งในโครงการนี้ได้นำเสนอ ความสามารถในการสื่อสารรับ-ส่งข้อมูลถึงกันผ่านตัวระบบควบคุมของรถ AGV ทั้ง 2 คัน และสามารถปรับส่งผลิตภัณฑ์ตามสถานีหรือจุดที่ต้องการได้ แบ่งหน้าที่หรืองานได้อย่างถูกต้องและทำงานได้อย่างปลอดภัยไม่เกิดอุบัติเหตุในสายการผลิต

คำสำคัญ : Automated Guided Vehicle (AGV)

Project Title Traffic Control System For AGV
Student: Mr. Sattawat Runganuwong
Department: Instrumentation and Control Engineering
Advisor: Assistant Professor Dr. Noppadol Maneerat
Mentor: Mr. Apiwat Ruengdit
Company: SCG Packaging Public Co., Ltd.

ABSTRACT

SCG Packaging Public Co., Ltd., which is engaged in the production of packaging. The assigned project is Traffic Control System For AGV. The company has the idea to reduce employees by using Automated Guided Vehicle (AGV) in the production line to replace the employees that moving products to conveyor. When working for a while, found that only 1 AGV is not enough to moving products to conveyor. If this problem is not resolved, this will lead more time to moving products. Therefore the company needs to increase AGV more than 1 vehicle by recognizing the problems that will follow. If don't have Traffic Control System For AGV, it could cause collisions among themselves or the employees in the production line. Therefore it come to be the idea to create this project. In which this project has presented the ability to communicate send and receive data through the control systems of both AGV vehicles and can transfer products to the desired station or point. Divide the duty or work correctly and work safely, without accidents in the production line.

Keyword : Automated Guided Vehicle (AGV)

กิตติกรรมประกาศ

โครงการเรื่อง ระบบควบคุมการวิ่งสำหรับรถ AGV สามารถสำเร็จลุล่วงไปด้วยดีเพราะได้รับความช่วยเหลือจากผู้ช่วยศาสตราจารย์ ดร.นพตล มณีรัตน์ ที่มีโอกาสในการเข้าร่วมโครงการสหกิจศึกษา ให้คำปรึกษา การสนับสนุน และความช่วยเหลือเป็นอย่างดี ตลอดจนการตรวจสอบความถูกต้องของรายงานจนทำให้รายงานฉบับนี้เสร็จสมบูรณ์ ผู้จัดทำโครงการรู้สึกซาบซึ้ง และขอกราบขอบพระคุณเป็นอย่างสูง อีกทั้งผู้จัดทำโครงการขอแสดงความขอบพระคุณต่อสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่มีโอกาสในการเข้าร่วมโครงการสหกิจศึกษา ทำให้ได้รับประสบการณ์ดี ๆ สามารถเป็นประโยชน์ในการทำงานในอนาคตได้

ผู้จัดทำโครงการขอขอบพระคุณบุคลากรทุกท่านของบริษัท เอสซีจี แพคเกจจิ้ง จำกัด (มหาชน) อาทิ นายอภิวัฒน์ เรืองดิษฐ์ ที่ให้การช่วยเหลือสนับสนุนในเรื่องต่างๆ เช่น การให้คำปรึกษา และให้ข้อมูลที่ เป็นประโยชน์ รวมทั้งทักษะในการแก้ไขปัญหา ดูแลติดตามความคืบหน้าของโครงการอยู่เสมอ ทำให้โครงการสามารถดำเนินไป และสำเร็จลุล่วงไปด้วยดี

ท้ายที่สุดนี้ขอขอบคุณผู้ที่เกี่ยวข้องที่ให้การช่วยเหลือทุกท่าน ที่ให้การช่วยเหลือมาโดยตลอด หากมีข้อผิดพลาดประการใด จึงขออภัยมา ณ ที่นี้

ศตวรรษ รุ่งอรุณวงศ์

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	II
สารบัญ.....	IV
สารบัญรูป.....	VI
สารบัญตาราง.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญ.....	1
1.2 วัตถุประสงค์ของโครงการ.....	2
1.3 ขอบเขตของโครงการ.....	2
1.4 วิธีการดำเนินโครงการ.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
บทที่ 2 ทฤษฎี และงานวิจัยที่เกี่ยวข้อง.....	4
2.1 ไมโครคอนโทรลเลอร์.....	4
2.2 โปรแกรม Microsoft Visual Studio.....	7
2.3 ภาษา C#.....	7
2.4 NodeMCU ESP8266.....	13
บทที่ 3 วิธีการดำเนินโครงการ.....	20
3.1 การวางแผนกระบวนการ และระยะเวลาในการดำเนินงาน.....	20
3.2 ศึกษาขั้นตอนการทำงานในสายการผลิต.....	21
3.3 ศึกษาวิธีรับ-ส่งข้อมูลระหว่าง Arduino และระบบควบคุม.....	22
3.4 ศึกษาการเขียนโปรแกรมด้วย Microsoft Visual Studio โดยใช้ภาษา C#.....	23
3.5 ออกแบบหน้า User Interface ของระบบควบคุม.....	24
3.6 ออกแบบกระบวนการทำงานของตัวระบบควบคุม.....	25
3.7 ออกแบบคำสั่งและเงื่อนไขในการทำงานของระบบควบคุม.....	27
3.8 ทดลองการทำงานของระบบควบคุม.....	29
3.9 แก้ไขข้อผิดพลาดของระบบควบคุม.....	29
บทที่ 4 ผลการดำเนินโครงการ.....	30
4.1 ผลการดำเนินโครงการ.....	30

สารบัญ(ต่อ)

หน้า

บทที่ 5 สรุปผลและข้อเสนอแนะ.....	31
5.1 สรุปผลการดำเนินโครงการ.....	31
5.2 ข้อเสนอแนะ.....	31
เอกสารอ้างอิง.....	33
ประวัติผู้เขียน.....	34



สารบัญรูป

รูปที่	หน้า
2.1 Arduino uno.....	4
2.2 โครงสร้าง Arduino Board.....	5
2.3 หน้าโปรแกรม Arduino IDE.....	6
2.4 หน้าต่างโปรแกรม Visual Studio.....	7
2.5 โครงสร้างโปรแกรมภาษา C# ขั้นพื้นฐาน.....	10
2.6 โครงสร้างโปรแกรมภาษา C# ขั้นพื้นฐาน กรณีไม่เขียนในส่วนของ namespace.....	11
2.7 ตัวอย่างโครงสร้างโปรแกรมภาษา C# ขั้นพื้นฐาน กรณีเขียนในส่วนของ namespace.....	12
2.8 ตัวอย่างโครงสร้างโปรแกรมภาษา C# ขั้นพื้นฐาน กรณีไม่เขียนในส่วนของ namespace.....	12
2.9 ESP8266 ESP-01.....	14
2.10 ESP8266 ESP-03.....	14
2.11 ESP8266 ESP-07.....	15
2.12 ESP8266 ESP-12E.....	15
2.13 ESP8266 NodeMCU V2.....	16
2.14 ESP8266 NodeMCU V3.....	17
2.15 WeMos D1.....	17
2.16 WeMos D1 Mini.....	18
2.17 Witty Cloud Mini NodeMCU.....	18
3.1 เครื่องผลิตกล่องกระดาษ.....	21
3.2 การตรวจสอบคุณภาพของกล่องกระดาษและการบรรจุลงท่อ.....	21
3.3 แผนผังสายการผลิต.....	22
3.4 หน้า User Interface ของระบบควบคุม.....	24
3.5 กระบวนการทำงานของระบบควบคุม.....	25
3.6 การทำงานโดยรวมของตัวระบบควบคุมในสายการผลิต.....	26
3.7 การเลือกลำดับ AGV ในการทำงานก่อนหรือหลัง.....	27
3.8 การเลือก AGV ในการหลบทางหรือไปต่อ.....	28
3.9 การนำทาง AGV จากจุดเริ่มต้นไปยังจุดหมาย.....	29

สารบัญตาราง

ตารางที่	หน้า
3.1 แผนการดำเนินงาน.....	20



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

บริษัท ปูนซีเมนต์ไทย จำกัด (มหาชน) หรือเครือซีเมนต์ไทย (Siam Cement Group : SCG) หรือที่นิยมเรียกว่า “ปูนใหญ่” ปัจจุบันใช้ชื่อเรียกทั้งกลุ่มธุรกิจว่า “เอสซีจี” เริ่มดำเนินการครั้งแรกเป็นผู้ผลิตปูนซีเมนต์และต่อมาได้มีการขยายกิจการไปสู่ผลิตภัณฑ์อื่นๆ อีกมากมาย อาทิเช่น ผลิตภัณฑ์ก่อสร้าง กระดาษ โลหะ เครื่องจักรกล ปีโตรเคมี ธุรกิจจัดจำหน่าย เป็นต้น มีสำนักงานทรัพย์สินส่วนพระมหากษัตริย์เป็นผู้ถือหุ้นรายใหญ่ประมาณ 30 เปอร์เซ็นต์ ที่เหลืออีก 70 เปอร์เซ็นต์ ถือหุ้นโดยนักลงทุนทั่วไป และนักลงทุนสถาบัน

ปัจจุบัน SCG มีบริษัทสำคัญกว่า 100 บริษัท มีพนักงานทั้งสิ้นประมาณ 28,000 คน ผลิตสินค้ากว่า 64,000 รายการ เพื่อจำหน่ายทั้งในประเทศและส่งออกไปยังทุกภูมิภาคทั่วโลก แต่เดิม SCG ดำเนินธุรกิจอยู่ในหลากหลายกลุ่มอุตสาหกรรม แต่หลังจากวิกฤตการณ์ทางการเงินในเอเชีย พ.ศ. 2540 SCG ได้มีการปรับโครงสร้างการบริหารและการดำเนินธุรกิจขนาดใหญ่ จนกระทั่งในปัจจุบัน SCG คงเหลือไว้ซึ่งกลุ่มธุรกิจที่สำคัญ 6 กลุ่มได้แก่ ซีเมนต์ ผลิตภัณฑ์ก่อสร้าง ปีโตรเคมี กระดาษและบรรจุภัณฑ์ ธุรกิจจำหน่าย และธุรกิจการลงทุน

เอสซีจี แพคเกจจิ้ง ได้เปลี่ยนชื่อจาก เอสซีจี เปเปอร์ เป็น เอสซีจีแพคเกจจิ้งตั้งแต่ปี 2558 เป็นต้นมา โดยการเปลี่ยนชื่อดังกล่าวสะท้อนถึงกลยุทธ์ของธุรกิจ ที่มุ่งเน้นด้านบรรจุภัณฑ์และยังสอดคล้องกับเป้าหมายการขยายธุรกิจในภูมิภาคเอเชียอีกด้วย ดังจะเห็นได้จากการขยายธุรกิจเข้าสู่ Flexible Packaging ในช่วง 2 ปีที่ผ่านมา นอกจากนี้ เอสซีจี แพคเกจจิ้ง ยังมุ่งเน้นการนำเสนอสินค้าและบริการด้านบรรจุภัณฑ์ ให้แก่ลูกค้าอย่างครบวงจรและยั่งยืน (Total Packaging Solutions Provider) ดังจะเห็นได้จากการพัฒนาสินค้าและบริการด้านบรรจุภัณฑ์ที่มีคุณภาพอย่างต่อเนื่อง

ในสายการผลิตบรรจุภัณฑ์อาหาร เมื่อสินค้าเสร็จสิ้นกระบวนการผลิตแล้วจะถูกนำมาบรรจุใส่ถุงพลาสติกใสและนำเข้ากล่องกระดาษลูกฟูก เพื่อเตรียมการจัดส่งให้กับลูกค้า ซึ่งในแต่ละวันต้องใช้พนักงานในการเดินขนย้ายสินค้าจากเครื่องผลิตไปที่จุดบรรจุใส่กล่องกระดาษลูกฟูก โดยทำให้งานมีประสิทธิภาพน้อยและเสียเวลามาก ซึ่งทำให้เสียงบประมาณในการฝึกอบรมให้พนักงานมีความชำนาญทางฝ่าย Engineering Design จึงมีความคิดในการแก้ปัญหาระยะยาว โดยการทำเครื่องจักรมาทำงานแทนพนักงาน จึงมีแนวคิดสร้างเครื่อง Automated Guided Vehicle (AGV) มาทำงานทดแทนพนักงานในการเคลื่อนย้ายสินค้า ซึ่งสามารถทำงานได้อย่างสม่ำเสมอ ลดอุบัติเหตุในการทำงานของพนักงานและนำพนักงานตรงส่วนนี้ไปทำในส่วนอื่นๆ ที่เครื่องจักรยังไม่สามารถทำได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อได้มีการนำ AGV มาใช้ในสายการผลิตแล้วทำให้พบปัญหาที่ว่า AGV เพียง 1 คันนั้นไม่เพียงพอต่อการเคลื่อนย้ายสินค้าให้หมดภายในการวิ่ง 1 รอบ ทำให้ต้องเพิ่ม AGV ขึ้นมาเพื่อแบ่งเบาภาระหรือช่วยกันทำงาน แต่เมื่อมี AGV มากกว่า 1 คันอยู่ในสายการผลิตแล้วนั้นทางฝ่าย Engineering Design จึงได้สังเกตเห็นถึงปัญหาที่จะตามมาหากไม่มีระบบควบคุมการทำงานของ AGV เพราะจะทำให้เกิดการชนกันเองหรือชนพนักงานในสายการผลิตได้ ดังนั้นจากปัญหาที่จะตามมาในอนาคตนี้ทำให้เกิดโครงการขึ้นนี้ขึ้นมาเพื่อช่วยควบคุมการทำงานของ AGV ที่มีอยู่ให้สามารถสื่อสารกันได้ แบ่งหน้าที่หรืองานได้อย่างถูกต้องและทำงานได้อย่างปลอดภัยไม่เกิดอุบัติเหตุในสายการผลิต

1.2 วัตถุประสงค์ของโครงการ

1. AGV ที่มีอยู่สามารถสื่อสารรับ-ส่งข้อมูลถึงกันได้ผ่านตัวระบบควบคุม
2. AGV แต่ละคันสามารถทำตามหน้าที่ที่ได้รับได้อย่างถูกต้อง
3. ลดจำนวนพนักงานในจุดปฏิบัติงาน

1.3 ขอบเขตของโครงการ

1. ออกแบบซอฟต์แวร์ระบบควบคุมผ่านโปรแกรม Microsoft Visual Studio 2019
2. เขียน Flow Chart กระบวนการทำงานและการทำตามเงื่อนไขต่างๆ ของระบบควบคุม
3. ศึกษาภาษา C# เพื่อใช้ในการเขียนคำสั่งและเงื่อนไขให้กับระบบควบคุม
4. ศึกษาวิธีการรับ - ส่งข้อมูลระหว่าง Arduino และระบบควบคุม

1.4 วิธีการดำเนินโครงการ

1. วางแผนในการดำเนินงาน
2. ศึกษาขั้นตอนการทำงานในสายการผลิต
3. ศึกษาการเขียนโปรแกรมด้วย Microsoft Visual Studio โดยใช้ภาษา C#
4. ศึกษาวิธีการรับ-ส่งข้อมูลระหว่าง Arduino และระบบควบคุม
5. ออกแบบหน้า User Interface ของตัวระบบควบคุม
6. ออกแบบกระบวนการทำงานของระบบควบคุม
7. ออกแบบคำสั่งและเงื่อนไขในการทำงานของระบบควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. ทดลองการทำงานของระบบควบคุม
9. แก้ไขข้อผิดพลาดของระบบควบคุม
10. จัดทำสรุปโครงการ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ประโยชน์ต่อตนเอง

การที่ได้เข้าร่วมโครงการสหกิจศึกษากับทางสถานประกอบการบริษัท บริษัท เอสซีจี แพคเกจจิ้ง จำกัด (มหาชน) ครั้งนี้นับเป็นประสบการณ์ที่ได้เรียนรู้ถึงการทำงานจริง การนำเอาสิ่งที่ได้เรียนรู้ภายในห้องเรียนมาประยุกต์ปรับใช้ในชีวิตการทำงาน รวมไปถึงได้เรียนรู้การทำงานที่เป็นระบบ การวางแผน และการเสริมสร้างทักษะในการทำงานต่างๆ การปรับตัวในสภาพแวดล้อมที่มีความกดดันที่เกิดขึ้นในระหว่างทำงาน

2. ประโยชน์ต่อสถานประกอบการ

การที่สถานประกอบการได้รับนักศึกษาเข้ามาปฏิบัติงานในโครงการสหกิจศึกษาโดยร่วมกับทางสถาบันฯ ทำให้ได้สร้างบุคลากรที่มีความสามารถในการทำงาน และเกิดการพัฒนากายในองค์กรของสถานประกอบการในอนาคต

3. ประโยชน์ต่อสถานศึกษา

การที่สถานศึกษาได้เปิดโครงการสหกิจศึกษา ทำให้มีการร่วมมือระหว่างตัวนักศึกษา อาจารย์ภาควิชาฯ และสถานประกอบการ ทำให้เกิดการพัฒนาในทุกๆ ฝ่ายภายในอนาคต และยังเป็น การตรวจสอบศักยภาพของสถานศึกษาผ่านตัวนักศึกษาที่ได้เข้าร่วมโครงการ ว่าผ่านการประเมิน มีความพร้อมที่จะทำงานมากน้อยแค่ไหน

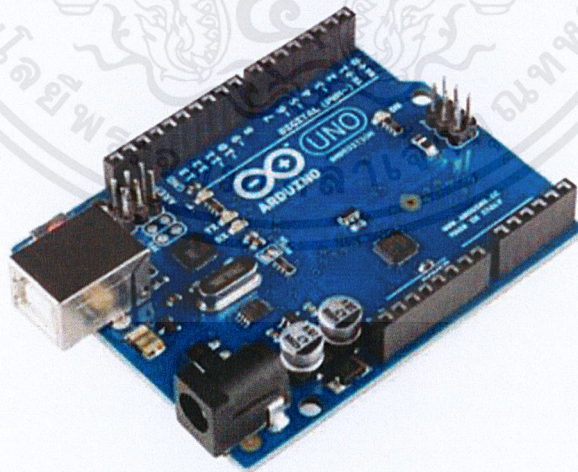
บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ไมโครคอนโทรลเลอร์

Arduino อ่านว่า (อา-ดู-อิ-โน้ หรือ อาดูยโน้) เป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือ มีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานง่ายและสะดวกในการใช้งาน โดยภาษาที่ใช้กับบอร์ดนี้จะมีลักษณะเป็น C/C++ โดยจัดให้มี Library ต่างๆ ให้พร้อมให้เรียกใช้งานได้ทันทีมากมาย ครอบคลุมการติดต่อกับ I/O ต่างๆ ได้หลากหลาย การใช้งานง่ายเพียงเสียบสาย USB ติดตั้งโปรแกรมที่ให้มาก็สามารถประยุกต์ใช้กับไมโครคอนโทรลเลอร์ได้ทันที ดังรูปที่ 2.1

ส่วนที่เป็น Hardware คือ บอร์ดอิเล็กทรอนิกส์ขนาดเล็กที่มีไมโครคอนโทรลเลอร์ (MCU) เป็นชิ้นส่วนหลัก ถูกนำมาประกอบร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่นๆ เพื่อให้ใช้งาน หรือที่เรียกกันว่า บอร์ด Arduino โดยบอร์ด Arduino เองก็มีหลายรุ่นให้เลือกใช้ตามสเปคที่ต้องการ ส่วนที่เป็น Software คือ ภาษา Arduino เป็นภาษาสำหรับเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ มีการเขียนแบบเดียวกับภาษา C/C++ และ Arduino IDE เป็นเครื่องมือสำหรับเขียนโปรแกรมด้วยภาษา Arduino, คอมไพล์โปรแกรม (Compile) และอัปโหลดโปรแกรมลงบอร์ด (Upload)

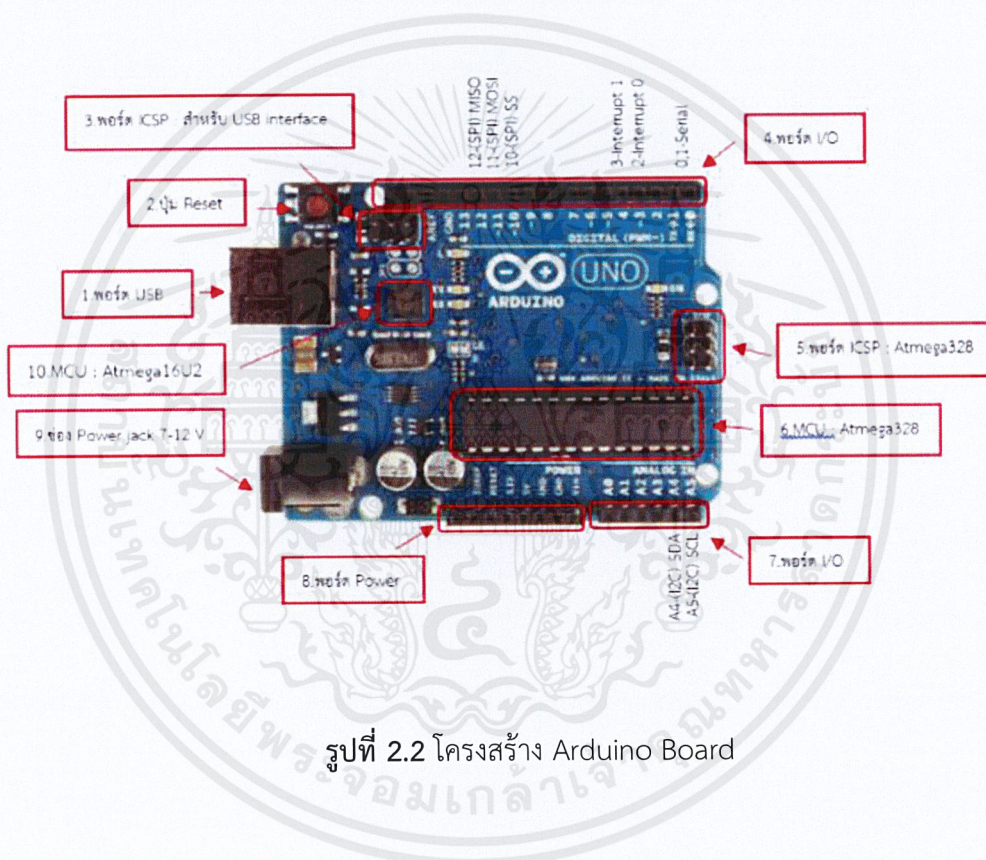


รูปที่ 2.1 Arduino UNO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บอร์ด Arduino มีโครงสร้างดังรูปที่ 2.2 และมีจุดเด่นคือ เรียนรู้และใช้งานง่าย เนื่องจากมีคำสั่งต่างๆ ขึ้นมาสนับสนุนในการใช้งานมีรูปแบบง่ายไม่ซับซ้อน ซึ่งถึงแม้ว่า Arduino เองจะมีรูปแบบการใช้งานคล้ายๆ กับไมโครคอนโทรลเลอร์อื่นๆ เช่น Basic Stamp, BX-24, Handy Board เป็นต้น แต่มีจุดเด่นที่ไม่เหมือนกับไมโครคอนโทรลเลอร์อื่นๆ อย่างเช่น

1. ราคาไม่แพง เนื่องจาก Source Code และวงจรแจกให้ฟรี สามารถต่อวงจรขึ้นเองได้
2. โปรแกรมที่ใช้พัฒนารองรับการทำงานทั้ง Windows, Linux และ Macintosh OSX
3. มีรูปแบบคำสั่งที่ง่ายต่อการใช้งาน แต่สามารถนำไปใช้กับงานจริงที่มีความซับซ้อนมากๆ ได้ และยังสามารถสร้างคำสั่งและ Library ใหม่ๆ มาใช้งานเองได้ เมื่อมีความชำนาญมากขึ้นแล้ว



รูปที่ 2.2 โครงสร้าง Arduino Board

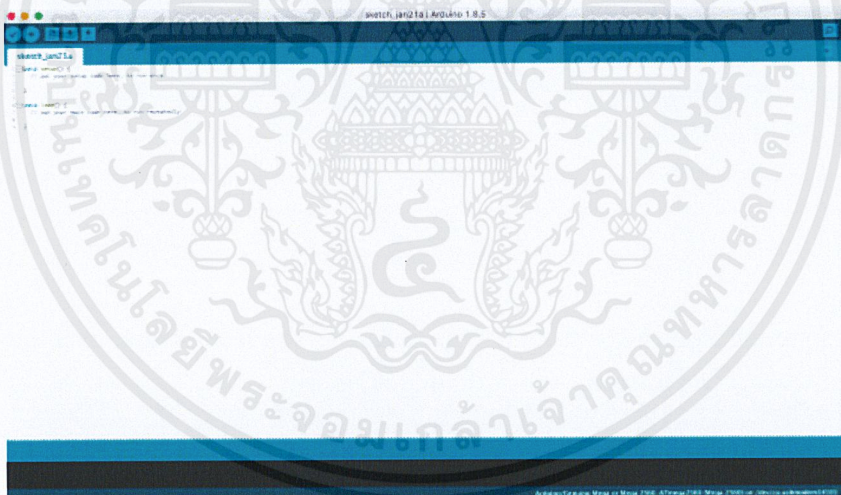
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 โปรแกรม Arduino IDE

Arduino IDE หรือ Arduino Integrated Development Environment คือ เครื่องมือที่ใช้สำหรับเขียนโปรแกรมควบคุมการทำงานของ Arduino โดยสามารถใช้งานได้กับ Arduino ทุกรุ่น โปรแกรม Arduino IDE เป็นโปรแกรม Open Source สามารถโหลดไปใช้งานได้ฟรี และยังมีโค้ดตัวอย่างให้ ทดสอบอุปกรณ์ต่างๆ เช่น โปรแกรมไฟกระพริบ โปรแกรมทดสอบมอเตอร์ เป็นต้น ดังรูปที่ 2.3

ขั้นตอนการใช้งาน Arduino IDE

1. ตั้งค่าบอร์ดให้ตรงกับบอร์ดที่ใช้งาน
2. ตั้งค่าพอร์ตเชื่อมต่อกับบอร์ด (กรณีต่อบอร์ดจริง)
3. ตั้งค่าชนิดการเขียนโปรแกรม (กรณีต่อบอร์ดจริง)
4. เขียนโปรแกรม
5. กดปุ่ม Compile เพื่อแปลงไฟล์เป็นภาษาเครื่อง
6. กดปุ่มเบิร์นไฟล์ลงบอร์ด (กรณีต่อบอร์ดจริง)

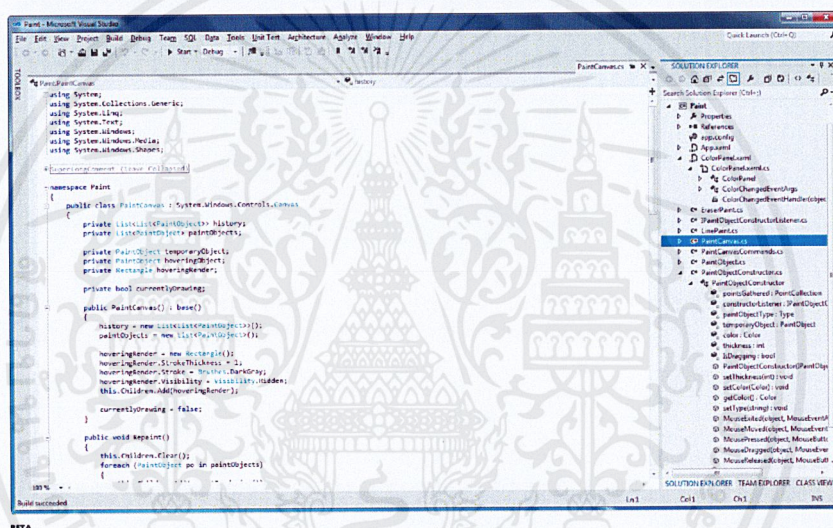


รูปที่ 2.3 หน้าโปรแกรม Arduino IDE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 โปรแกรม Microsoft Visual Studio

Visual Studio คือ โปรแกรมตัวหนึ่งที่เป็นเครื่องมือที่ช่วยพัฒนาซอฟต์แวร์และระบบต่างๆ ซึ่งสามารถติดต่อสื่อสารพูดคุยกับคอมพิวเตอร์ได้ในระดับหนึ่งแล้ว แต่ยังไม่สามารถพัฒนาเป็นระบบเองได้เหมาะสมสำหรับภาษา VB และ VB.NET เนื่องจากไมโครซอฟต์ได้พัฒนาโปรแกรมและภาษาขึ้นมาควบคู่กันเพื่อให้ใช้งานได้ง่ายซึ่งกันและกัน ซึ่งนักโปรแกรมเมอร์จะนำเครื่องมือมาใช้ในการพัฒนาต่อยอดให้เกิดเป็นระบบต่างๆ หรือเป็นเว็บไซต์ และแอปพลิเคชันต่างๆ ซึ่งเป็นเครื่องมือที่ใช้สร้างโปรแกรมต่างๆ เช่น โปรแกรมที่รันบนระบบปฏิบัติการ Windows เช่น โปรแกรมคำนวณเลข โปรแกรมฐานข้อมูล เช่น Microsoft Access, Microsoft SQL Server คอมพิวเตอร์ทางด้าน Active X โปรแกรมที่รันบนอินเทอร์เน็ต ดังรูปที่ 2.4



รูปที่ 2.4 หน้าต่างโปรแกรม Visual Studio

2.3 ภาษา C#

ภาษา C# จะเขียนว่า C Sharp ก็ได้ (อ่านว่าซีชาร์ป) เครื่องหมาย # ในทางดนตรีหมายถึงครึ่งเสียง ส่วนคำว่า Sharp หมายถึงเฉียบคม มีไหวพริบ บริษัทไมโครซอฟท์ตั้งชื่อภาษานี้ว่า C# เพื่อจะสื่อให้เห็นว่าเป็นภาษาในตระกูลภาษา C เช่นเดียวกับภาษา C++ คือ เป็นภาษาที่มีต้นกำเนิดจากภาษา C หากท่านสังเกตให้ดีจะเห็นว่าสัญลักษณ์ # จะมองให้เป็นเครื่องหมาย + สื่อนำมาซ้อนกันก็ได้ แสดงเป็นนัยว่า ภาษานี้ก้าวหน้ากว่าภาษา C++ ไปอีกระดับหนึ่ง (คือเป็นภาษา C+++) สัญลักษณ์ # นี้มีปรากฏในภาษาอื่นๆ ของ .NET ด้วย เช่นภาษา J# (เดิมคือภาษา J++) และภาษา A# (เหมือนภาษา Ada) อาจเป็นเพื่อเหตุผลทางการการค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาษา C# เป็นภาษาที่ใหม่มากๆ ปรากฏตัวเป็นครั้งแรกในปี 2000 และถูกอัปเดตเวอร์ชันอยู่เสมอ มันเป็นภาษาที่ได้รับอิทธิพลจากภาษาก่อนหน้าเช่นภาษา Delphi ภาษา C++ ภาษา Java และภาษา Eiffel ในตอนต้นภาษานี้ถูกออกแบบ และกำหนดลักษณะโดยบริษัทไมโครซอฟท์ ต่อมาได้ถูกรับรองจากหน่วยงาน ECMA (หน่วยงานกำหนดมาตรฐานสากลด้านสารสนเทศ) และ ISO แต่ปัจจุบันไมโครซอฟท์ยังพัฒนาภาษานี้อย่างต่อเนื่อง (ปัจจุบันเป็นเวอร์ชัน 3.0) ภาษา C# ถูกพัฒนาขึ้นโดยเป็นส่วนหนึ่งในการพัฒนาโครงสร้างพื้นฐานของ .NET Framework เป็นการกรณนำข้อดีของภาษาต่างๆ (เช่นภาษา Delphi ภาษา C++) มาปรับปรุงเพื่อให้มีความเป็น OOP อย่างถึงที่สุด ขณะเดียวกันก็ลดความซับซ้อนในโครงสร้างของภาษาลง (เรียบง่ายกว่าภาษา C++) และมีเครื่องแต่งตัวน้อยลง (เมื่อเทียบกับ Java)

2.3.1 ประวัติภาษา C#

ภาษา C# เป็นภาษา โปรแกรมเชิงวัตถุทำงานบน .Net framework พัฒนาโดยบริษัทไมโครซอฟท์ และมี Anders Hejlsberg เป็นหัวหน้าโครงการ โดยมีรากฐานมาจากภาษา C++ และภาษาอื่นๆ (โดยเฉพาะภาษาเดลไฟและจาวา) โดยปัจจุบันภาษา C# เป็นภาษามาตรฐานรองรับโดย ECMA และ ISO ซึ่งในปัจจุบันได้พัฒนาและปรับรูปแบบของ ภาษา C# อยู่ตลอดเวลาโดยทาง Microsoft ได้นำภาษา C# ไปอยู่ในชุดพัฒนา Software อย่าง Visual Studio ซึ่งทำให้เป็นที่นิยมเพิ่มมากขึ้น ไมโครซอฟท์ส่งมาตรฐาน ภาษา C# ให้กับ ECMA และได้รับการยอมรับเป็นมาตรฐาน ECMA ในเดือนธันวาคม ค.ศ.2001 ในชื่อว่า ECMA-334 C# Language Specification ใน ค.ศ. 2003 ภาษา C# ได้รับการยอมรับเป็นมาตรฐาน ISO (ISO/IEC 23270) มาตรฐาน ISO/IEC 23270:2003 ระบุรูปแบบ และกำหนดการแปล (ตีความ) โปรแกรมที่เขียนด้วยภาษา C# โดยตัวมาตรฐานได้ระบุ

- รูปแบบการนำเสนอ
- ไวยากรณ์
- กฎการตีความสำหรับแปลโปรแกรมภาษา C#
- ข้อห้าม และข้อจำกัดของเครื่องมือที่สร้างตามข้อกำหนดของ C# ISO/IEC 23270:2003 ไม่ได้

ระบุ

- กลไกในการแปลงโปรแกรมภาษา C# เพื่อใช้ในระบบประมวลผลข้อมูล
- กลไกในการเรียกให้โปรแกรมภาษา C# ทำงาน เพื่อใช้ในระบบประมวลผลข้อมูล
- กลไกในการแปลงข้อมูลเข้า เพื่อใช้กับโปรแกรมภาษา C#
- กลไกในการแปลงข้อมูลออก หลังจากถูกประมวลผลโดยโปรแกรมภาษา C#

นอกจากนี้ตัวมาตรฐานไม่ได้กล่าวถึง โครงสร้างข้อมูล (Data Structure) และตัว Library กลางของ .NET Framework ซึ่งเป็นหัวใจสำคัญของการเขียนโปรแกรมด้วยภาษา C# เลย

2.3.2 จุดเด่นของ C#

เป็นรูปแบบของภาษาที่ทำงานเป็นลำดับ (Sequential) และต้องผ่านการแปลงไฟล์ให้อยู่ในรูปแบบพร้อมใช้งานหรือ EXE file (ที่เรียกว่าการ Compile) เช่นเดียวกับภาษา C โดย C# เป็นภาษาที่ถูกผลักดัน โดยบริษัทไมโครซอฟท์ให้ออกมายิ่งใหญ่โดยมีจุดเด่นดังนี้

- เขียนคล่อง C# มีเครื่องมือในการช่วยเขียนอย่าง Visual Studio ที่มีพีเจอร์ในการไฮไลต์โค้ด และส่วนช่วยในการเขียนโปรแกรมที่สะดวกมาก
- เขียนง่ายจุดเด่นของตัวแปรและอ็อบเจกต์บน C# เด่นชัดในเรื่องของการจัดการคุณสมบัติ (Properties) และการตั้งค่าเริ่มต้นที่ช่วยให้สามารถพัฒนาระบบได้สะดวกรวดเร็วขึ้น
- อ่านง่าย การจัดระเบียบโดยตัว Visual Studio จะจัดการกับย่อหน้าทั้งหมดเหล่านีให้อยู่ในระเบียบสะอาดตาที่สุด
- เป็นอ็อบเจกต์ เนื่องจากภาษา C# นั้นมีแม่แบบมาจากภาษา Java ซึ่งจุดเด่นคือ การทำทุกสิ่งให้เป็นวัตถุ (Object) ทำให้สามารถเขียนและพัฒนาได้ง่าย
- ประสิทธิภาพสูง C# เป็นภาษาที่พัฒนาขึ้นมาภายใต้ .NET Framework ซึ่งสามารถดึงเอาความสามารถของเทคโนโลยีบน .NET ออกมาใช้ได้อย่างเต็มประสิทธิภาพ
- สามารถทำงานระดับลึก ภาษา C# สามารถทำงานกับหน่วยความจำรวมถึงระบบคอมพิวเตอร์ในระดับลึก โดยผ่านพอยเตอร์ (Pointer) หรือทำงานกับโปรโตคอล TCP/IP ที่ต่ำกว่าระดับ 4 ได้
- เน้นที่ XML C# ออกแบบมาเพื่อใช้งานร่วมกับ XML ได้อย่างราบรื่นที่สุดด้วยความช่วยเหลือของ NET Framework
- คุณสมบัติ Generic Type ช่วยยืดหยุ่นในการประกาศตัวแปร
- เทคโนโลยี LINQ ช่วยในการทำงานกับฐานข้อมูลได้อย่างมีประสิทธิภาพ

2.3.3 เป้าหมายการใช้งาน

ภาษา C# เป็นภาษาที่มีต้นกำเนิดมาจากภาษา C เช่นเดียวกับ C++ และถูกพัฒนาขึ้นโดยเป็นส่วนหนึ่งในการพัฒนาโครงสร้างพื้นฐานของ .NET Framework ซึ่งเป็นการนำข้อดีของภาษาต่างๆ เช่น Delphi, C++ มาปรับปรุงเพื่อให้มีความเป็น OOP อย่างถึงที่สุด ขณะเดียวกันก็พัฒนาให้มีความเรียบง่ายกว่าภาษา C++ นอกจากนี้ ภาษา C# ใช้การเขียนโปรแกรมที่ซับซ้อน เช่น โปรแกรมควบคุมฮาร์ดแวร์ และโปรแกรมเกมสามมิติ เพราะ .NET จะมีภาษาให้เลือกใช้หลายสิบภาษา แต่ภาษา C# เป็นพื้นเมืองของ .NET เมื่อค้นใน web จะพบข้อมูลและตัวอย่างโค้ดเป็นภาษา C# มากกว่าภาษาอื่น

2.3.4 สถานภาพในปัจจุบัน

ปัจจุบันภาษา C# ยังคงใช้งานกันอยู่และเป็นที่ยอมรับกันมาก เช่น ใช้ในการเรียนการสอน ใช้ในการเขียนโปรแกรมเกม หรือใช้ในการเขียนโปรแกรมต่างๆ

เนื่องจาก C# มีต้นกำเนิดมาจาก C นั้นเอง เครื่องหมาย # เป็นสิ่งที่แสดงถึงความก้าวหน้ากว่า C++ ไปอีกระดับหนึ่ง C# ได้รวบรวมข้อดีของภาษาต่างๆ เช่น Java, Delphi, C++ เข้าไว้ด้วยกัน อีกทั้งยังมีความเรียบง่ายกว่า อีกทั้งยังมีเครื่องมือดีๆ อย่าง Visual C# 2008 ของทางไมโครซอฟท์อีกด้วย และยังมี การรวมเอาความสามารถของภาษา Visual C++ มารวมกับความใช้งานง่ายของภาษา Visual Basic ทำให้ ภาษา Visual C# เป็นภาษาคอมพิวเตอร์ที่มีความสามารถสูงในขณะเดียวกันใช้งานง่าย ทำให้การพัฒนา โปรแกรมทำได้อย่างสะดวก ง่ายตาย และรวดเร็ว

2.3.5 โครงสร้างของภาษา C#

โครงสร้างโปรแกรมภาษา C# ชั้นพื้นฐานจะประกอบด้วยส่วนของโปรแกรมหลัก แต่จะไม่มีส่วนของโปรแกรมย่อย (Subroutine) โดยแสดงดังรูปที่ 2.5

```

namespace (1)
{
    class (2)
    {
        static void Main ( )
        {
            (3)
        }
    }
}

```

รูปที่ 2.5 โครงสร้างโปรแกรมภาษา C# ชั้นพื้นฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.5 แสดงโครงสร้างโปรแกรมภาษา C# ขั้นพื้นฐาน โดยมีรายละเอียดดังนี้

1. หมายเลข (1) เป็นการระบุชื่อของ namespace ซึ่งใช้ในการกำหนดขอบเขตให้กับคลาสต่างๆ รวมถึงใช้ในการจัดโครงสร้างของโปรแกรมขนาดใหญ่ให้เป็นสัดส่วนอีกด้วย โดยเฉพาะอย่างยิ่งในการเขียนโปรแกรมคอมพิวเตอร์ที่ซับซ้อนโดยมีผู้เขียนโปรแกรมหลายคน นอกจากนี้การกำหนด namespace ยังช่วยป้องกันปัญหาการตั้งชื่อคลาสหรือค่าคงที่อื่นๆ ซ้ำกันได้
2. หมายเลข (2) เป็นการระบุชื่อของ class
3. หมายเลข (3) เป็นการระบุพื้นที่สำหรับคำสั่งต่างๆ ที่ผู้เขียนโปรแกรมต้องการให้คอมพิวเตอร์ปฏิบัติตาม

นอกจากนี้ ในบางกรณีผู้เขียนโปรแกรมสามารถที่จะไม่เขียนในส่วนของ namespace ได้ ถ้าโปรแกรมคอมพิวเตอร์นั้นมีขนาดเล็ก และไม่ซับซ้อนมากนัก ซึ่งการที่ไม่เขียนในส่วนของ namespace จะถือว่า class ที่ถูกสร้างขึ้นมาอยู่ใน namespace กลาง โดยแสดงดังรูปที่ 2.6

```

class (2)
{
    static void Main ( )
    {
        (3)
    }
}

```

รูปที่ 2.6 โครงสร้างโปรแกรมภาษา C# ขั้นพื้นฐาน กรณีไม่เขียนในส่วนของ namespace

ตัวอย่าง โครงสร้างโปรแกรมภาษา C# ขั้นพื้นฐาน

ซึ่งจะแสดงข้อความ Hello C# ออกทางจอภาพ และจากนั้นรอนกว่าผู้ใช้งานจะกด Enter แล้วจบการทำงาน

กรณีที่ 1 เขียนในส่วนของ namespace โดยแสดงดังรูปที่ 2.7

```
namespace HelloApp
{
    class HelloC#
    {
        static void Main ( )
        {
            System.Console.WriteLine ("Hello C#");
            System.Console.ReadLine ( );
        }
    }
}
```

รูปที่ 2.7 ตัวอย่างโครงสร้างโปรแกรมภาษา C# ขั้นพื้นฐาน กรณีเขียนในส่วนของ namespace

กรณีที่ 2 ไม่เขียนในส่วนของ namespace โดยแสดงดังรูปที่ 2.8

```
class HelloC#
{
    static void Main ( )
    {
        System.Console.WriteLine ("Hello C#");
        System.Console.ReadLine ( );
    }
}
```

รูปที่ 2.8 ตัวอย่างโครงสร้างโปรแกรมภาษา C# ขั้นพื้นฐาน กรณีไม่เขียนในส่วนของ namespace

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 NodeMCU ESP8266

ESP8266 เป็นชื่อเรียกของชิพของโมดูล ESP8266 สำหรับติดต่อสื่อสารบนมาตรฐาน WiFi ทำงานที่แรงดันไฟฟ้า 3.0-3.6 โวลต์ ทำงานใช้กระแสโดยเฉลี่ย 80 มิลลิแอมป์ รองรับคำสั่ง Deep Sleep ในการประหยัดพลังงาน ใช้กระแสต่ำกว่า 10 ไมโครแอมป์ สามารถ Wake Up กลับมาส่งข้อมูลใช้เวลาต่ำกว่า 2 มิลลิวินาที ภายในมี Low Power MCU 32 บิต ทำให้เราเขียนโปรแกรมสั่งงานได้ มีวงจร Analog Digital Converter ทำให้สามารถอ่านค่าจาก Analog ได้ความละเอียด 10 บิต ทำงานได้ที่อุณหภูมิ -40 ถึง 125 องศาเซลเซียส เมื่อนำชิพ ESP8266 มาผลิตเป็นโมดูลหลายรุ่น ก็จะขึ้นต้นด้วย ESP8266 แล้วตามด้วยรุ่น เช่น ESP-01, ESP-03, ESP-07, ESP-12E

ESP8266 ติดต่อกับ WI-FI แบบ Serial สามารถเขียนโปรแกรมลงไปในชิพ โดยใช้ Arduino IDE ได้ ทำให้การเขียนโปรแกรมและใช้งานเป็นเรื่องง่าย คล้ายกับการใช้ Arduino แน่ใจว่าสามารถติดต่ออุปกรณ์อื่นๆ เช่น เซอร์ต่างๆ แบบสไตล์ Arduino ถ้ามีพื้นฐาน Arduino อยู่แล้ว ก็เข้าใจและใช้งานได้รวดเร็ว

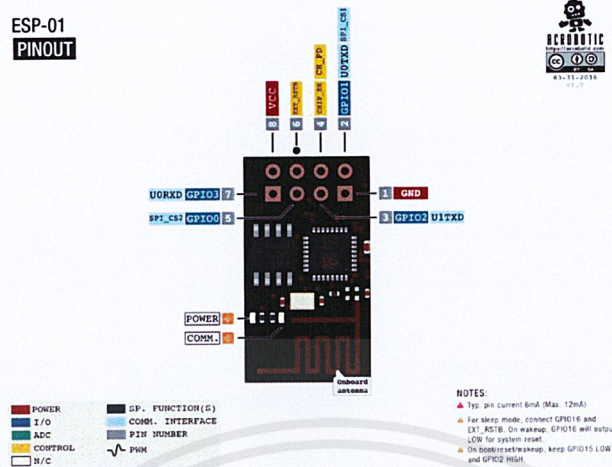
โมดูล ESP8266 มีหลายรุ่น และมีรุ่นใหม่พัฒนาออกมาเรื่อยๆ โดยโครงสร้างและขาที่ใช้งานก็จะมีลักษณะคล้ายกันคือ

- GPIO0 เป็นขาสำหรับเลือกโหมด โดยเมื่อต่อกับ GND จะเข้าโหมดโปรแกรม เมื่อต้องการให้ทำงานปกติก็ไม่ต้องต่อ
- GPIO15 เป็นขาที่ต้องต่อลง GND เพื่อให้โมดูลทำงาน
- CH_PD หรือ EN เป็นขาที่ต้องต่อไฟ VCC เพื่อ Pull Up สัญญาณ ให้โมดูลทำงาน โมดูลบางรุ่นไม่มีขา Reset มาให้ เมื่อต้องการรีเซ็ต ให้ต่อขา CH_PD กับ GND
- Reset ต่อกับไฟ VCC เพื่อ Pull Up สัญญาณ โดยเมื่อต้องการรีเซ็ต ให้ต่อกับไฟ GND
- VCC เป็นขาสำหรับจ่ายไฟเลี้ยง ใช้ไฟเลี้ยง 3.0-3.6 โวลต์
- GND ต่อกับไฟ 0 โวลต์
- GPIO เป็นขาดิจิตอล INPUT/OUTPUT ทำงานที่ไฟ 3.3 โวลต์
- ADC เป็นขา Analog INPUT รับแรงดันสูงสุด 1 โวลต์ ความละเอียด 10 บิต หรือ 1024 ค่า

เวลาโปรแกรมเพียงมองหาขาเหล่านี้ แล้วต่อให้ครบเท่าที่มีขาให้ต่อก็สามารถโปรแกรม ESP8266 ได้ทุกรุ่น

โมดูล ESP8266 รุ่นที่นิยมเช่น ESP-01, ESP-03, ESP-07, ESP-12E นอกจากนี้ยังมีบอร์ด ESP8266 ที่รวมวงจร USB TTL เข้าไปทำให้โปรแกรมกับ Arduino ได้ง่ายขึ้นเช่น NodeMCU, Wemos D1, Wemos mini

ESP8266 ESP-01



รูปที่ 2.9 ESP8266 ESP-01

ESP8266 ESP-01 เป็นรุ่นที่มีขาต่อออกมาข้างนอก จึงสามารถต่อกับบอร์ดทดลองได้ง่าย มีขา GPIO 4 ขา คือ GPIO0, GPIO1, GPIO2, GPIO3

ESP8266 ESP-03

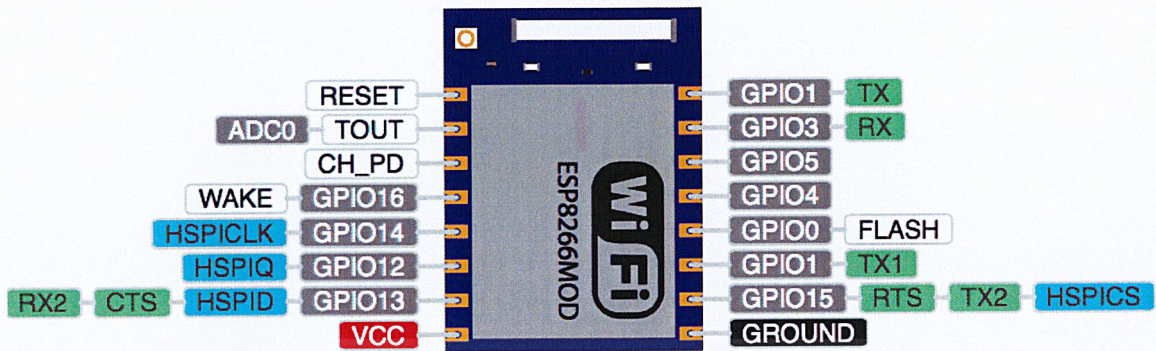


รูปที่ 2.10 ESP8266 ESP-03

ESP8266 ESP-03 จะคล้ายกับ ESP-01 มี Package การต่อขาเป็นแบบเซอร์เฟซเมาส์ (Surface Mount) โดยจะมีขา GPIO มากขึ้น ทำให้สั่งงานอุปกรณ์ได้มากกว่า ESP8266 ESP-01 โดย ESP8266-13 มีขาทั้งหมด 14 ขา มีเสาอากาศแบบมาให้อยู่ในตัว และยังสามารถต่อสายอากาศเพิ่มเพื่อเพิ่มกำลังการรับส่งได้ที่ขา 14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

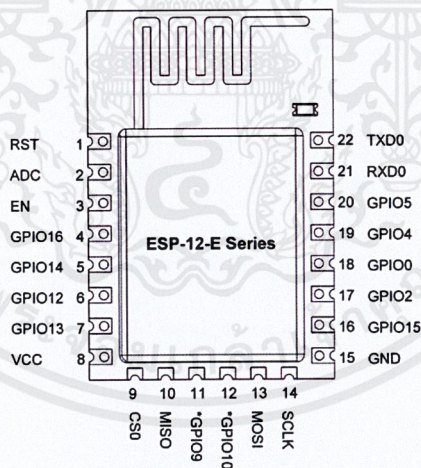
ESP8266 ESP-07



รูปที่ 2.11 ESP8266 ESP-07

ESP8266 ESP-07 เพิ่มแผ่นเหล็กครอบชิพ ESP8266 ไว้เพื่อป้องกันสัญญาณรบกวน มีขาทั้งหมด 16 ขา โดยมีขา GPIO ให้ใช้งาน 11 ขาและขา Analog Read อีก 1 ขา

ESP8266 ESP-12E



รูปที่ 2.12 ESP8266 ESP-12E

ESP8266 ESP-12E มีการจัดขาและต่อใช้งานแบบเดียวกับรุ่น ESP-07 โดยเปลี่ยนเสาอากาศมาเป็นแบบเดินวงจรภายใน PCB และเพิ่มขาอีก 6 ขา คือ SCLK MOSI MISO สำหรับติดต่อกับเซนเซอร์อื่นๆ โพรโตคอล SPI ซึ่งรุ่นนี้ก็ได้รับความนิยมเป็นอย่างมาก จนเกิดการต่อยอดเป็นบอร์ด ESP8266-12 รุ่นต่างๆ

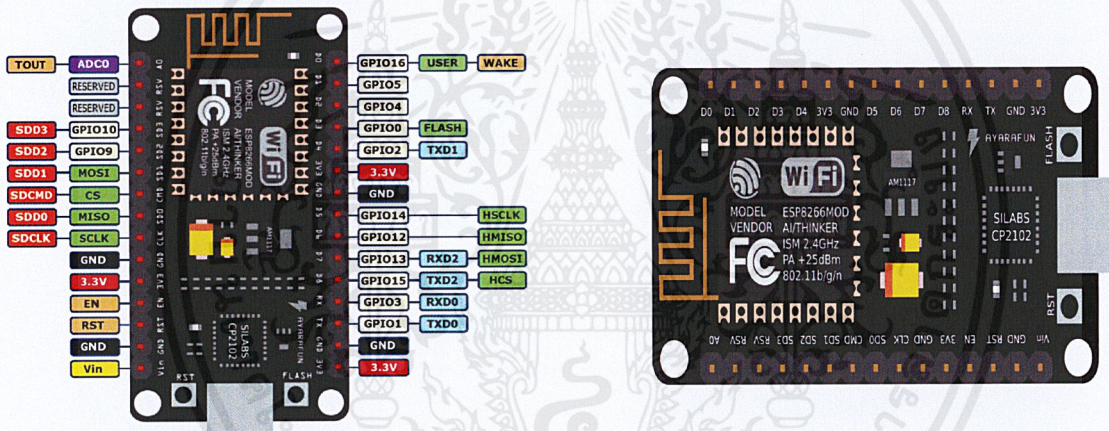
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 การเขียนโปรแกรม ESP8266 Arduino IDE

สำหรับการต่อวงจรของ ESP8266 เพื่อเขียนโปรแกรมด้วย Arduino เขียนโค้ดเหมือนกันทุกรุ่น โดยแต่ละรุ่นจะมีขาไม่เท่ากัน ดังนั้นจึงต้องเปรียบเทียบขา GPIO ให้ถูกต้องในการใช้งานก็ใช้ได้แล้ว การเขียนโปรแกรมอัปโหลดโค้ดลงบอร์ด ESP8266 เกือบทุกรุ่น จะผ่านทาง Serial ที่ขา RX, TX โดยใช้โมดูล USB TTL ซึ่งต้องเสียเวลาในการต่อวงจรเพื่ออัปโหลดโค้ด อีกทั้งโมดูล ESP8266 หลายๆ รุ่นมีการต่อขาที่เป็นแบบเซอร์เฟซเมสส์ (Surface Mount) ทำให้ไม่สะดวกกับการต่อทดลองบนบอร์ดทดลอง

ดังนั้นจึงมีการรวม โมดูล USB TTL และต่อวงจรขยายขา ESP8266 ให้เป็นขาระยะห่างขนาด 2.54 มิลลิเมตร ซึ่งสามารถเสียบลงบอร์ดทดลองได้พอดี กลายเป็น บอร์ด ESP8266 โดยหนึ่งในบอร์ดที่นิยมใช้งานคือ NodeMCU ซึ่งใช้ โมดูล ESP8266 ESP-12E ดังรูปที่ 2.13

บอร์ด ESP8266 NodeMCU V2

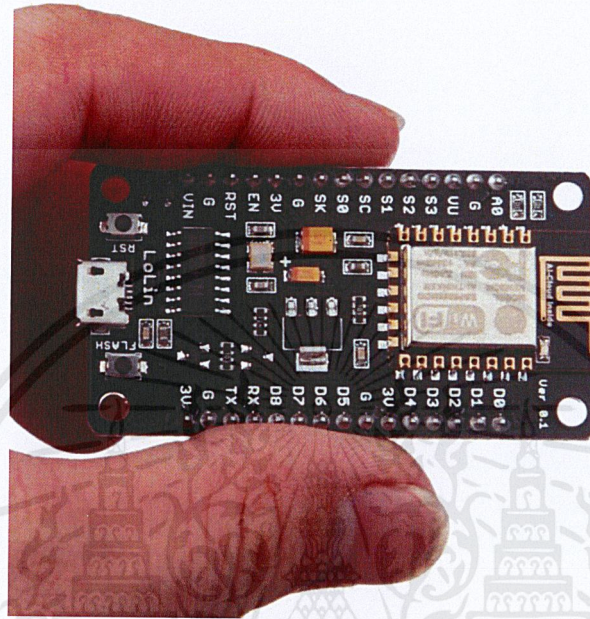


รูปที่ 2.13 บอร์ด ESP8266 NodeMCU V2

NodeMCU V2 เป็น ESP8266-12E รวมกับ USB TTL ที่ใช้ชิพ CP2102 และขยายขาให้สามารถต่อทดลองได้ง่ายขึ้น มีปุ่ม Reset และ Flash สำหรับใช้โปรแกรม โดยใช้ Arduino IDE หรือโปรแกรมอื่นๆ ได้อย่างสะดวก

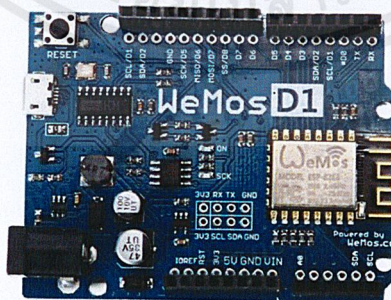
บอร์ด ESP8266 NodeMCU V3

NodeMCU V3 เป็นบอร์ดที่คล้ายกับ NodeMCU V2 ที่ต่างกันคือ NodeMCU V3 จะมีขนาดกว้างกว่า และใช้ชิพ USB TTL เป็น CH340 ซึ่งการต่อขาใช้งานและโค้ดโปรแกรมเหมือนกันทุกประการ ดังรูปที่ 2.14



รูปที่ 2.14 บอร์ด ESP8266 NodeMCU V3

บอร์ด WeMos D1

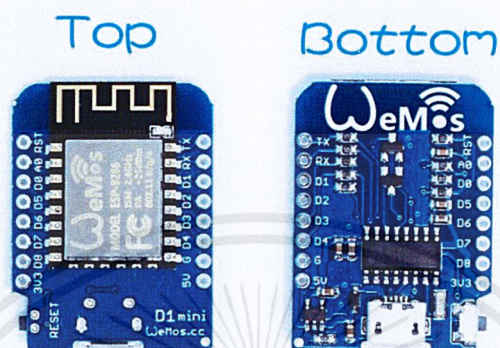


รูปที่ 2.15 บอร์ด WeMos D1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกแบบให้คล้ายบอร์ด Arduino Uno ใช้ ESP8266-12E เป็นตัวหลัก ชิพ CH340 เป็นภาค USB TTL รองรับไฟ 9-12 โวลต์ ที่ช่อง Adapter, รองรับไฟ 5 โวลต์ ที่ช่อง micro usb และ 3.3 โวลต์ ที่ช่อง VIN

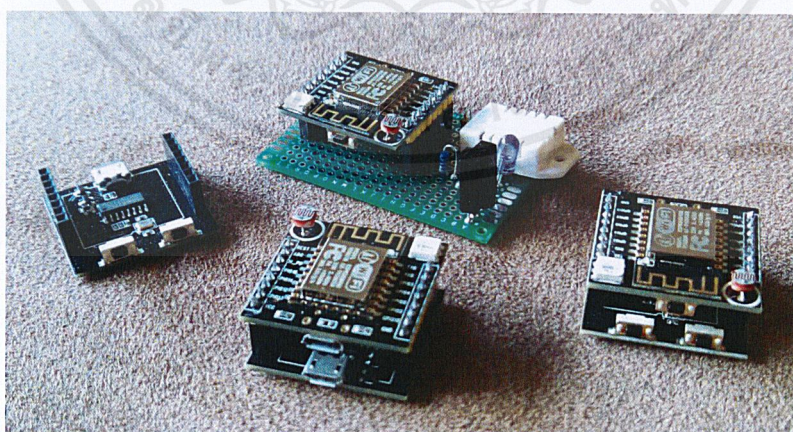
บอร์ด WeMos D1 Mini



รูปที่ 2.16 บอร์ด WeMos D1 Mini

แปลงบอร์ด Wemos D1 ให้มีขนาดเล็กลง โดยการแยกส่วน USB TTL และ ESP8266-12E ซอดี คือ เวลาใช้งานจริงสามารถถอดเฉพาะส่วน ESP8266-12E ไปใช้งานก็จะทำให้ประหยัดไฟได้นิดหน่อย

Witty Cloud Mini NodeMCU



รูปที่ 2.17 บอร์ด Witty Cloud Mini NodeMCU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บอร์ดนี้มีขนาดเล็กแยกส่วนได้คล้ายๆ กับ WeMos D1 Mini โดยใช้ ESP8266-12F โดยมี LDR, สวิตช์ และ LED RGB ต่อมาให้ในบอร์ดทำให้สะดวกในการต่อทดลอง ข้อดีของรุ่นนี้คือ ฐานออกแบบมาให้มาขนาดเท่ากับบอร์ดขยายขาของ ESP-07, ESP-08, ESP-12 ทำให้สามารถนำไปใช้โปรแกรมกับ ESP รุ่นดังกล่าวได้ด้วย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

วิธีการดำเนินโครงการ

ในการดำเนินโครงการนี้ ทางบริษัทได้มอบหมายโครงการให้เขียนซอฟต์แวร์เพื่อใช้ควบคุมการวิ่งรถ AGV หรือเป็นระบบควบคุมการวิ่งสำหรับรถ AGV โดยขั้นตอนการจัดทำโครงการนี้นั้นจะเริ่มตั้งแต่การศึกษาขั้นตอนในสายการผลิต การวางแผนในการดำเนินงาน การศึกษาหาข้อมูลที่เกี่ยวข้องกับโครงการ การออกแบบตัวระบบควบคุม ไปจนถึงการทดลองตัวระบบควบคุมว่าสามารถใช้งานได้จริงหรือไม่

3.1 การวางแผนกระบวนการ และระยะเวลาในการดำเนินงาน

ในการดำเนินโครงการจะต้องมีการวางแผนเป็นขั้นตอน จัดการเวลาในการทำงานให้เหมาะสม เพื่อให้สามารถดำเนินโครงการให้เสร็จตามเวลาที่ได้รับมอบหมาย โดยระยะเวลาที่ดำเนินโครงการนั้นเป็นช่วงเวลาในโครงการสหกิจศึกษาที่บริษัทเอสซีจีแพคเกจจิ้ง จำกัด (มหาชน) ช่วงระยะเวลาระหว่างวันที่ 3 มิถุนายน พ.ศ. 2562 ถึงวันที่ 29 พฤศจิกายน พ.ศ. 2562 แสดงแผนการดำเนินงานดังตารางที่ 3.1

ตารางที่ 3.1 แผนการดำเนินงาน

รายละเอียด	มิถุนายน				กรกฎาคม				สิงหาคม				กันยายน				ตุลาคม				พฤศจิกายน			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
ศึกษาขั้นตอนการทำงานในสายผลิต																								
วางแผนการดำเนินงาน																								
ศึกษาและรวบรวมข้อมูล																								
ออกแบบตัวระบบควบคุม																								
เขียนคำสั่งและเงื่อนไขให้กับตัวระบบควบคุม																								
ทดลองตัวระบบควบคุม																								
สรุปผลการทดลองและทำรูปเล่มรายงาน																								

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ศึกษาขั้นตอนการทำงานในสายการผลิต

ในอุตสาหกรรมขนาดใหญ่ การลำเลียงผลิตภัณฑ์ให้มีความปลอดภัยนั้นเป็นสิ่งสำคัญ หากเกิดอุบัติเหตุระหว่างการขนส่งอาจส่งผลให้เกิดความล่าช้า ความเสียหายของเสียผลิตภัณฑ์ ความสูญเสียของชีวิตพนักงานและชื่อเสียงของโรงงานได้

ลักษณะการทำงานภายในสายการผลิต ในโรงงานกล่องกระดาษ Fest Hub (เป็นบริษัทในเครือของบริษัท เอสซีจีแพคเกจจิ้ง จำกัด (มหาชน)) เป็นการทำงานที่พนักงาน 1 คนจะรับผิดชอบเครื่องป้อนกล่อง กระดาษ 2 เครื่อง โดยพนักงานจะนำผลิตภัณฑ์ที่เครื่องทำเสร็จมาตรวจสอบงานและบรรจุลงห่อ เมื่อลงห่อเสร็จแล้ว จะมีพนักงานนำรถเข็นมารับห่อผลิตภัณฑ์ไปบรรจุลงกล่อง โดยโครงการนี้ทำขึ้นเพื่ออำนวยความสะดวกและลดภาระงานของพนักงาน ดังรูปที่ 3.1 ถึงรูปที่ 3.3

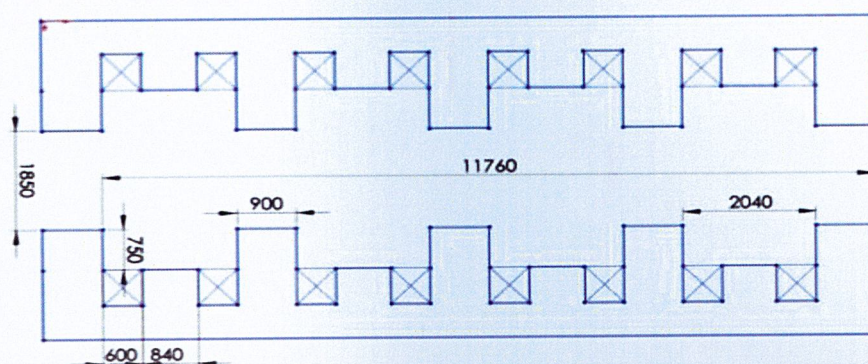


รูปที่ 3.1 เครื่องผลิตกล่องกระดาษ



รูปที่ 3.2 การตรวจสอบคุณภาพของกล่องกระดาษและการบรรจุลงห่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แผนผังสายการผลิต

3.3 ศึกษาวิธีรับ-ส่งข้อมูลระหว่าง Arduino และระบบควบคุม

เมื่อศึกษาแล้วจึงได้ทราบว่าผู้ที่ทำงานกับโครงข่ายคอมพิวเตอร์ และต้องทำงานกับไอพีแอดเดรส จะรู้จักโปรโตคอลสองตัวที่มักคุ้นเสมอคือ TCP (Transmission Control Protocol) และ UDP (User Datagram Protocol)

TCP และ UDP เป็นโปรโตคอลในชั้น Transport Layer ซึ่งสามารถชี้ IP (Internet Protocol) หรือหมายความว่าทั้งสองตัวสามารถใช้ไอพีแอดเดรส ในการระบุคอมพิวเตอร์ปลายทางได้เหมือนกัน ความแตกต่างที่เห็นได้ชัดระหว่าง TCP และ UDP คือ Connection, State และ Sequence ของทั้งสองตัว โดย Connection หมายถึงการเชื่อมต่อ ในที่นี้คือ การทำให้แน่ใจว่าผู้ส่งสารและผู้รับสารมีตัวตนและสามารถติดต่อถึงกันได้อย่างแน่นอน ส่วน State หมายถึงสถานะ ในที่นี้คือ โปรโตคอลจะต้องมีความสามารถในการรู้ถึงสถานะของสารที่ส่งไป เช่น สารถึงมือผู้รับหรือยัง หรือสารส่งไปยังผู้รับสารได้ถูกต้องครบถ้วนหรือไม่ และ Sequence หมายถึงการเรียงลำดับสาร ในที่นี้คือ การทำให้แน่ใจว่าสารที่ส่งและรับจะต้องเรียงลำดับกันได้อย่างถูกต้อง จะต้องทำให้แน่ใจว่าผู้รับสารจะต้องได้รับสารลำดับที่ 1 ก่อนลำดับที่ 2 เสมอ

TCP มีครบทั้งสามอย่าง ผู้ส่งสารและผู้รับสารต้องมีการส่งข้อมูลทักทายกันไปมาหรือที่เรียกว่าแฮนด์เชค (Handshake) อย่างน้อยสามครั้งก่อนส่งสาร เพื่อให้แน่ใจผู้ส่งสารและผู้รับสารมีตัวตนอยู่จริง รวมถึงมีการกำหนดไทม์เอาต์ (Timeout) ของสารทุกสารที่ส่งออกไป หากผู้ส่งสารไม่ได้รับการตอบสนองจากผู้รับสารตามเวลาที่กำหนดไว้ในไทม์เอาต์ ผู้ส่งสารจะต้องเริ่มส่งสารใหม่ นอกจากนั้นสารทุกสารยังมีการกำหนดลำดับ เพื่อให้ผู้รับสารแน่ใจได้ว่าไม่มีสารที่ตกหล่นหรือทำซ้ำอย่างแน่นอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

UDP ไม่มีทั้งสามอย่างที่กล่าวไว้ข้างต้น โดยถูกออกแบบมาให้เป็นโปรโตคอลที่เรียบง่ายที่สุดใน การส่งสารจากผู้ส่งไปยังผู้รับ โดยใช้เวลาน้อยที่สุด ด้วยเหตุนี้จึงไม่มีการรับประกันว่าสารที่ส่งไปจะถึงผู้รับ รวมถึงไม่รับประกันว่าข้อมูลส่งออกไปจะครบถ้วน การตรวจสอบความถูกต้องหรือความครบถ้วนของข้อมูล จะถูกยกให้เป็นหน้าที่ของแอปพลิเคชัน (Application) ของทั้งผู้ส่งและผู้รับ

อาจจะสรุปได้ว่าแบบ TCP จะได้รับความถูกต้องมากกว่าแต่จะเขียนได้ช้า ตรงข้ามกับการทำงาน แบบ UDP ซึ่งจะได้ความไวแต่อาจจะมีข้อความตกหล่นไปบ้าง ด้วยเหตุนี้ TCP จึงถูกใช้ในแอปพลิเคชัน ทั่วไปที่ทำงานแบบคู่โคลอนท์-เซิร์ฟเวอร์หรือยูนิคแอสต์ (Unicast : ผู้ส่งสารหนึ่ง-ผู้รับสารหนึ่งแบบระบุตัว) และต้องการความถูกต้องของข้อมูลเป็นหลัก เช่น แอปพลิเคชันรับส่งเมล แอปพลิเคชันฐานข้อมูล ในขณะที่ UDP ใช้ในแอปพลิเคชันที่ต้องการการส่งข้อมูลอย่างต่อเนื่องและว่องไว เช่น แอปพลิเคชันดูหนังฟังเพลง ออนไลน์ หรือเกมออนไลน์ หรือบิททอเรนท (Bit Torrent) เป็นต้น รวมไปถึงแอปพลิเคชันที่มีลักษณะ ผู้รับสารหลายตัวพร้อมกันทั้ง มัลติแอสต์ (Multicast : ผู้ส่งสารหนึ่งตัว-ผู้รับสารหลายตัวแบบระบุตัว) และ บรอสแอสต์ (Broadcast : ผู้ส่งสารหนึ่งตัว-ผู้รับสารหลายตัวแบบไม่ระบุตัว) เป็นต้น

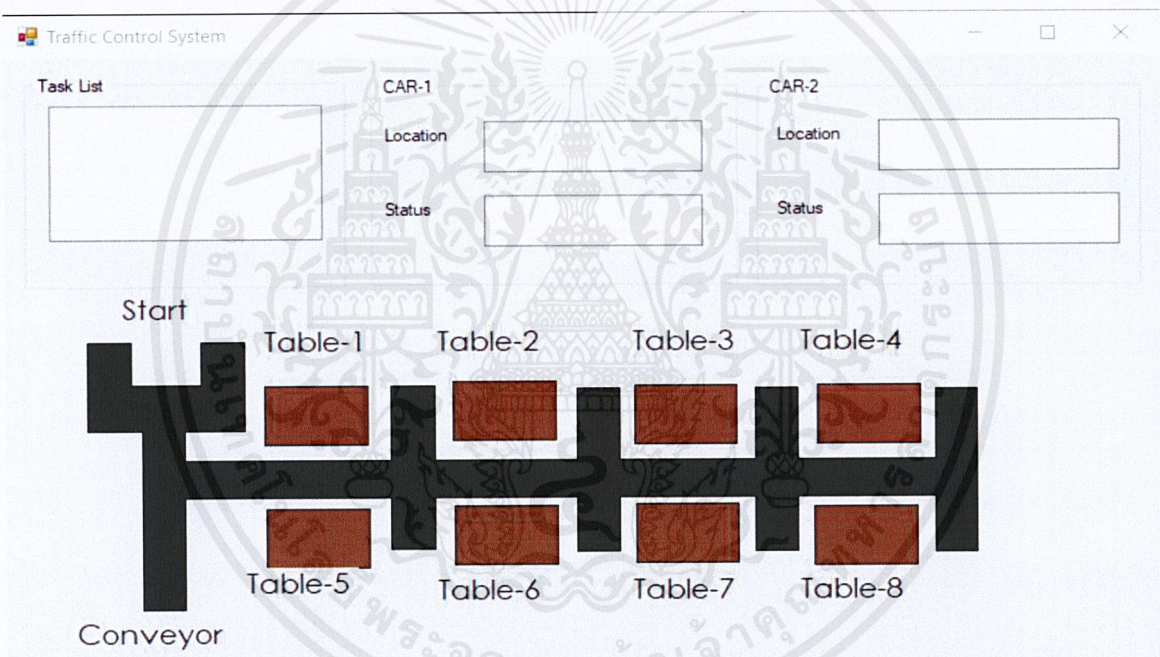
เนื่องจากต้องการให้มี AGV หรือผู้รับสารหลายคันในระบบจึงเลือกใช้วิธีแบบ UDP ซึ่งมีความ รวดเร็วและสามารถส่งให้ผู้รับหลายตัวแบบระบุตัวหรือระบุเลข IP ได้ ซึ่งเป็นสิ่งที่ TCP ไม่สามารถทำได้ ส่วนเรื่องข้อเสียของ UDP ที่อาจจะมีข้อความตกหล่นนั้นจะให้เป็นที่หน้าที่ของตัวระบบควบคุมและตัว Arduino เป็นตัวตรวจเช็คว่าคุณสมบัติที่ได้รับมานั้นครบถ้วนหรือไม่

3.4 ศึกษาการเขียนโปรแกรมด้วย Microsoft Visual Studio โดยใช้ภาษา C#

ศึกษาความแตกต่างระหว่างภาษา C# กับภาษา C ซึ่งภาษา C# มีความแตกต่างกับภาษา C ใน หลายๆ ด้านจึงต้องศึกษาเพิ่มเติม และศึกษาวิธีสร้างโปรแกรมนามสกุล .exe ด้วย Microsoft Visual Studio เพื่อที่จะสามารถเขียนตัวระบบควบคุมแล้วนำไปใช้ในระบบปฏิบัติการ Windows หรือ ระบบปฏิบัติการอื่นที่รองรับได้

3.5 ออกแบบหน้า User Interface ของระบบควบคุม

ในการออกแบบหน้า UI (User Interface) ได้ออกแบบโดยใช้เครื่องมือที่มีอยู่ให้ในโปรแกรม Microsoft Visual Studio มาใช้ในการออกแบบ โดยจะประกอบไปด้วยส่วนที่แสดงถึงคำสั่งที่ได้รับมาจากตัวพนักงานที่ประจำในแต่ละโต๊ะ, สถานะของ AGV แต่ละคัน (ในที่นี่ได้ทำการออกแบบไว้สำหรับ AGV เพียง 2 ตัวเนื่องจากการเริ่มต้นพัฒนาระบบควบคุมขึ้นมา จึงเริ่มที่จะควบคุมเพียง 2 ตัวก่อนเมื่อสามารถควบคุม 2 ตัวได้อย่างมีประสิทธิภาพจึงจะพัฒนาต่อไป) และตำแหน่งของ AGV ว่าอยู่ในที่ตำแหน่งใดในสายการผลิต ซึ่งถ้าต้องการที่จะเพิ่มหรือแก้ไขหน้า UI ให้มีรายละเอียดหรือแสดงค่าต่างๆ เพิ่มขึ้นตามต้องการ นั้นต้องเข้าไปทำการแก้ไขโดยโปรแกรม Microsoft Visual Studio อย่างเดียว ซึ่งก็จำเป็นที่จะต้องเพิ่มหรือแก้ไขโค้ดที่เกี่ยวข้องกับส่วนๆ นั้นด้วย โดยหน้า User Interface ที่ได้ออกแบบมานั้นแสดงดังรูปที่ 3.4

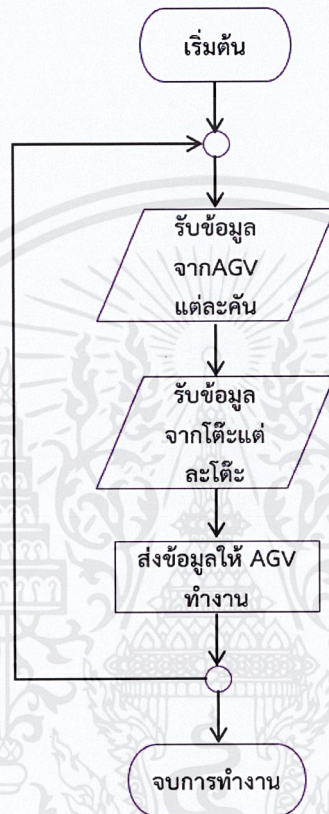


รูปที่ 3.4 หน้า User Interface ของระบบควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 ออกแบบกระบวนการทำงานของตัวระบบควบคุม

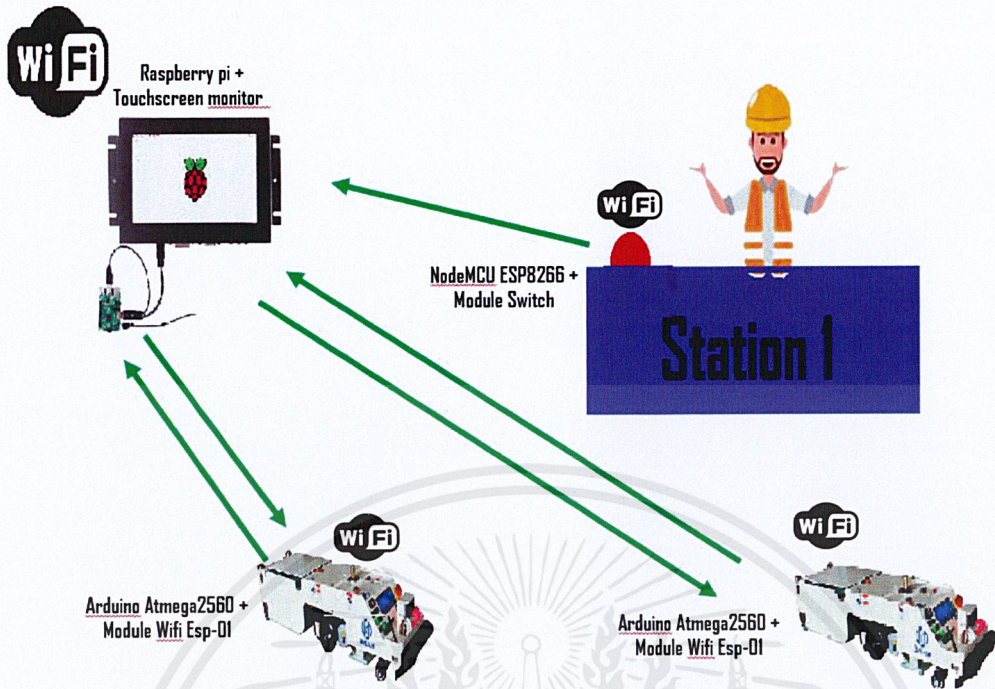
ก่อนที่จะเขียนเงื่อนไขหรือคำสั่งให้กับตัวระบบควบคุม ก็ต้องทำการออกแบบกระบวนการทำงาน โดยการเขียนเป็น Flow Chart การทำงานของระบบควบคุม ทั้งนี้เพื่อทำให้เกิดความเข้าใจที่ง่ายและตรงกันของผู้ที่เขียนเงื่อนไขให้กับตัวระบบควบคุม ซึ่งจะทำให้ผู้ที่มาพัฒนาระบบควบคุมต่อจากผู้พัฒนาคนก่อนจะสามารถพัฒนาได้ตรงจุด และไม่สับสนในโค้ดหรือเงื่อนไขที่อยู่ในตัวระบบก่อนหน้านั้น ดังแสดงในรูปที่ 3.5



รูปที่ 3.5 กระบวนการทำงานของระบบควบคุม

เนื่องจาก AGV แต่เดิมนั้นใช้ Arduino Atmega2560 เป็นตัวคอนโทรลเลอร์หลัก ซึ่งไม่สามารถที่จะเชื่อมต่อ WiFi ได้จึงต้องทำการต่อ Module WiFi Esp-01 เข้าไปในวงจรเพื่อที่จะสามารถส่ง-รับข้อมูลกับระบบควบคุมผ่าน WiFi ได้ ในส่วนบนโต๊ะของพนักงานที่ทำการบรรจุกล่องกระดาษลงท่อนั้นจะมีปุ่มหรือ Module Switch ไว้สำหรับเรียก AGV เพื่อมารับ-ส่งสินค้าโดย Module Switch จะต่อเข้ากับ NodeMCU ESP8266 V3 ซึ่งการเขียนโปรแกรมจะเหมือนกับตัว Arduino แต่สามารถเชื่อมต่อ WiFi ได้ไม่จำเป็นต้องใช้ Module WiFi มาต่อเพิ่ม โดยการทำงานโดยรวมของระบบควบคุมในสายการผลิตแสดงดังรูปที่ 3.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

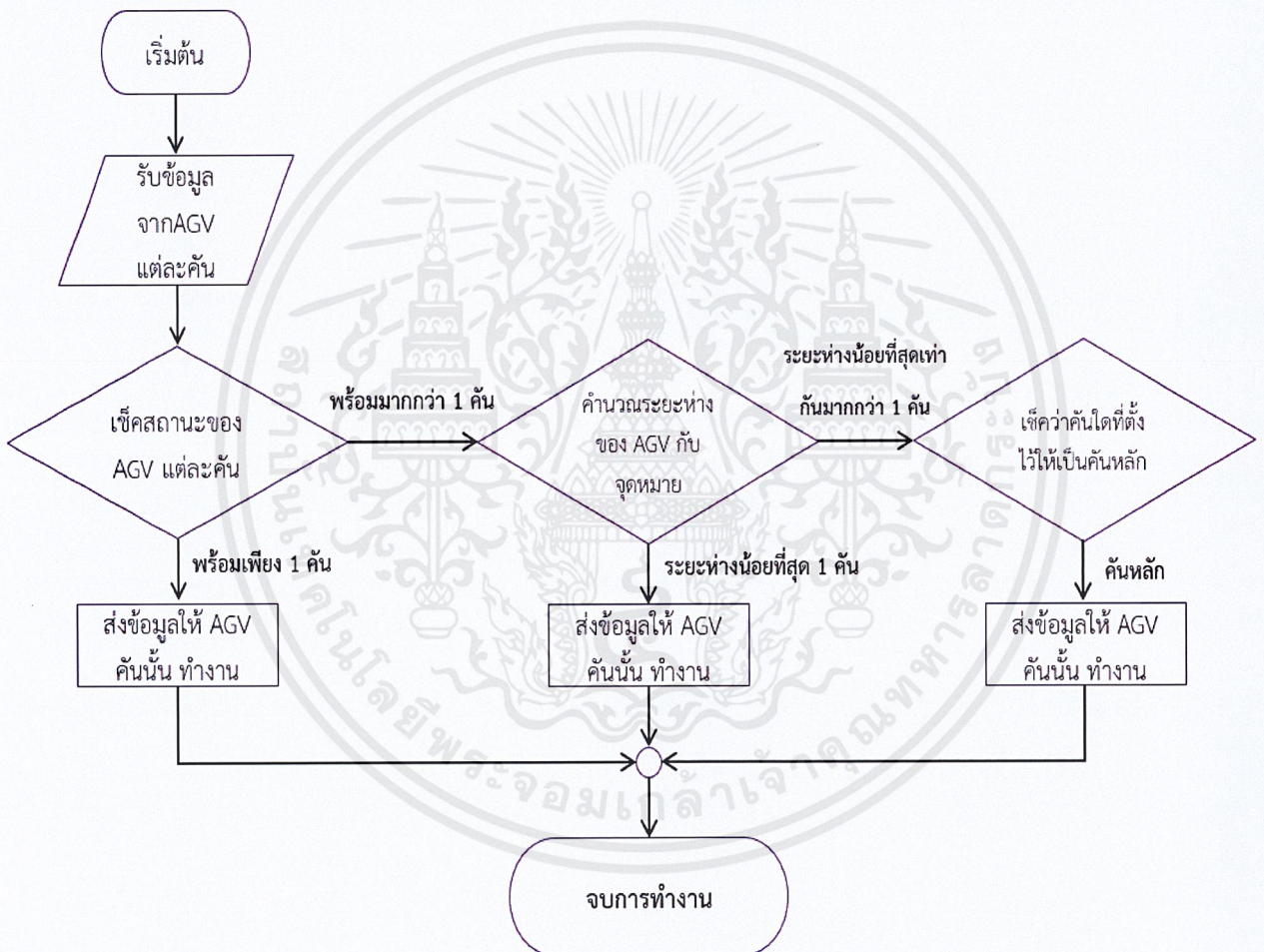


รูปที่ 3.6 การทำงานโดยรวมของตัวระบบควบคุมในสายการผลิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

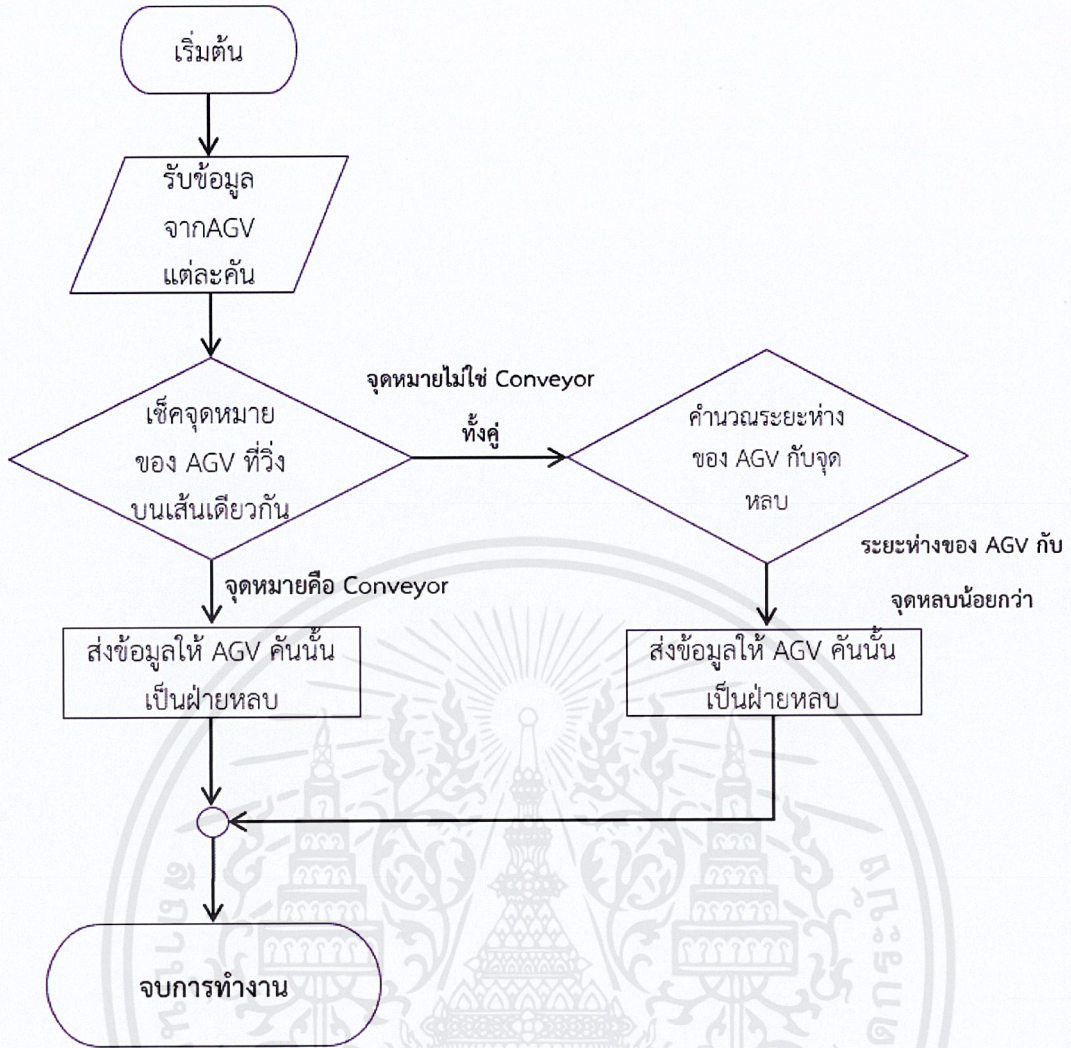
3.7 ออกแบบคำสั่งและเงื่อนไขในการทำงานของระบบควบคุม

เมื่อระบบควบคุมมีหลักการทำงานแล้วขั้นต่อไปจึงเป็นส่วนที่สำคัญ เพราะเป็นส่วนที่จะทำให้ตัวระบบควบคุมสามารถทำงานได้เอง และควบคุม AGV ได้ตามที่คุณพัฒนาที่กำหนดไว้ ซึ่งเงื่อนไขที่จะนำมาใช้ในระบบควบคุมขึ้นอยู่กับผู้พัฒนาว่าต้องการความซับซ้อนหรือความหลากหลายมากเพียงใด โดยในที่นี้ได้เลือกเงื่อนไขที่จะนำไปใช้ในระบบควบคุมดังนี้ 1. การเลือกลำดับ AGV ในการทำงานก่อนหรือหลัง 2. การเลือก AGV ในการหลบทางหรือไปต่อ 3. การนำทาง AGV จากจุดเริ่มต้นไปยังจุดหมาย ซึ่งจาก 3 เงื่อนไขนี้จะสามารถเขียนเป็น Flow Chart เพื่อให้ผู้พัฒนาสามารถเข้าใจอย่างง่ายได้ดังรูปที่ 3.7, รูปที่ 3.8 และรูปที่ 3.9



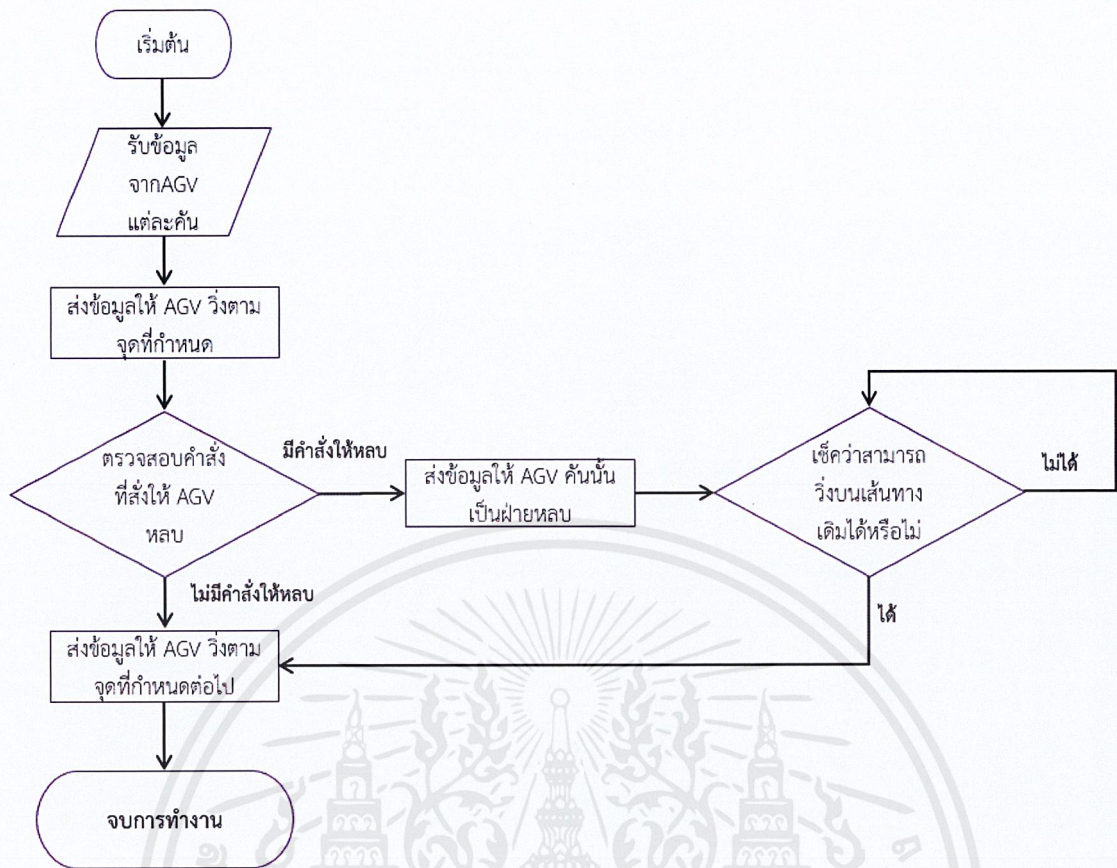
รูปที่ 3.7 การเลือกลำดับ AGV ในการทำงานก่อนหรือหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 การเลือก AGV ในการหลบทางหรือไปต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 การนำทาง AGV จากจุดเริ่มต้นไปยังจุดหมาย

3.8 ทดลองการทำงานของระบบควบคุม

เมื่อระบบควบคุมที่ออกแบบเขียนเงื่อนไขเสร็จสมบูรณ์จึงนำไปทดลองในการควบคุม AGV เพื่อตรวจสอบหาข้อผิดพลาด ซึ่งอาจจะเกิดมาจากเงื่อนไขที่เขียนในโปรแกรม Microsoft Visual Studio มีการตกหล่นหรือไม่ครบถ้วน

3.9 แก้ไขข้อผิดพลาดของระบบควบคุม

เมื่อทดลองจนรู้ถึงข้อผิดพลาดแล้ว จากนั้นจึงทำการวิเคราะห์หาสาเหตุของความผิดพลาดที่เกิดขึ้น แล้วจึงแก้ไขต่อไป ซึ่งเมื่อแก้ไขไปแล้วก็อาจจะพบข้อผิดพลาดอื่นๆ เกิดขึ้นได้อีกในภายหลัง เพราะข้อผิดพลาดบางกรณีอาจจะไม่สามารถทดลองแล้วเจอข้อผิดพลาดนั้นได้ทันที ดังนั้นจึงควรทำให้ระบบมีข้อผิดพลาดน้อยที่สุดเท่าที่จะทำได้

บทที่ 4

ผลการดำเนินโครงการ

4.1 ผลการดำเนินโครงการ

สำหรับผลการดำเนินโครงการจะกล่าวถึงระบบควบคุมที่ได้ออกแบบไว้ หลังจากการศึกษาวิธีการรับ-ส่งข้อมูลระหว่าง Arduino กับระบบควบคุม รวมไปถึงศึกษาเรื่องการเขียนโปรแกรมด้วย Microsoft Visual Studio โดยใช้ภาษา C# เมื่อทำการเขียนคำสั่งและเงื่อนไขตามที่ออกแบบไว้ให้กับระบบควบคุมแล้วนั้นระบบควบคุมสามารถรับ-ส่งข้อมูลให้กับ AGV ตามที่ต้องการได้ อีกทั้งระบบควบคุมยังสามารถแสดงค่าสถานะและตำแหน่งของ AGV แต่ละตัวได้อย่างถูกต้อง แต่ก็มีข้อผิดพลาดที่สำคัญก็คือ ระบบควบคุมยังไม่สามารถมอบหรือกำหนดหน้าที่ให้กับ AGV แต่ละตัวได้อย่างถูกต้อง หรือยังไม่สามารถทำงานได้ทุกเงื่อนไขตามที่ต้องการ ซึ่งข้อผิดพลาดนี้จะทำให้ ระบบควบคุมยังไม่สามารถนำไปใช้ในสายการผลิตได้จริงเพราะว่าอาจจะเกิดอุบัติเหตุต่อพนักงานในสายการผลิต รวมถึงเกิดความเสียหายแก่ผลิตภัณฑ์ได้



บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผลการดำเนินโครงการ

โครงการนี้ได้นำเสนอเกี่ยวกับระบบควบคุมการวิ่งสำหรับ AGV ซึ่งทำหน้าที่ในการทำให้ AGV สามารถสื่อสารรับ-ส่งข้อมูลถึงกันได้ รวมไปถึงการควบคุมการทำงานของ AGV ให้ทำตามหน้าที่ได้อย่างถูกต้อง ซึ่งผลการดำเนินงานนั้นอยู่ในระดับพอใช้ เนื่องจากระบบควบคุมยังไม่สามารถใช้งานจริงได้ถึงแม้ว่าระบบควบคุมจะสามารถรับ-ส่งข้อมูลให้แก่ AGV และยังสามารถแสดงค่าสถานะและตำแหน่งของ AGV ได้จริง แต่ก็ยังไม่สามารถทำให้ระบบควบคุมสามารถมอบหรือกำหนดหน้าที่ให้ AGV ทำตามได้อย่างถูกต้อง ซึ่งผิดจากจุดประสงค์ที่ตั้งไว้ โดยมีสาเหตุมาจากคำสั่งและเงื่อนไขที่เขียนให้กับระบบควบคุมนั้นไม่ครบถ้วน และยังไม่ถูกต้อง รวมไปถึงยังมีความรู้และความชำนาญที่ไม่มากพอที่จะทำให้ระบบควบคุมสำเร็จได้ตามจุดประสงค์ที่ตั้งไว้ ทำให้เมื่อนำไปใช้จริงจะทำให้เกิดการดำเนินงานที่ผิดพลาดของ AGV ได้ซึ่งจะเกิดผลกระทบที่ตามมาหากการนำไปใช้ในสายการผลิต เช่น เกิดอุบัติเหตุต่อพนักงาน, AGV วิ่งชนกันเอง, AGV ไม่สามารถทำตามที่ต้องการได้ทำให้เสียเวลาในการขนย้ายสินค้า เป็นต้น

5.2 ข้อเสนอแนะ

1. ควรสื่อสารรายละเอียดโครงการให้เข้าใจกับพี่ที่ปรึกษา หากสงสัย หรือไม่มั่นใจในงานที่ได้รับมอบหมายต้องสอบถามให้เข้าใจ และชัดเจน เพื่อไม่ให้เกิดการทำงานที่ผิดพลาด ซึ่งจะส่งผลให้เกิดปัญหาตามมาในภายหลัง
2. ควรจะมีการวางแผนในการออกแบบ รวมไปถึงวางแผนในการดำเนินให้ดี ต้องมีการเผื่อระยะเวลาไว้สำหรับปัญหาหรือข้อผิดพลาดที่จะเกิดขึ้น ซึ่งถ้ามีการเผื่อเวลาไว้จะทำให้สามารถแก้ปัญหาได้ทันตามกำหนดการ
3. ผู้ที่พัฒนาโครงการเรื่องนี้ต่อหากสามารถทำให้ระบบควบคุมการวิ่งรถ AGV สมบูรณ์ได้แล้วนั้น ในกรณีนี้ทำการออกแบบไว้สำหรับ AGV 2 ตัว อยากให้พัฒนาระบบควบคุมการวิ่งสำหรับ AGV ที่มากกว่า 2 คันขึ้นไปให้สมบูรณ์ด้วยเพื่อที่จะได้เป็นระบบควบคุมการวิ่งสำหรับ AGV ที่สามารถใช้งานได้จริงกับทุกสายการผลิตและสามารถลดจำนวนพนักงานได้มากยิ่งขึ้น
4. ความรู้พื้นฐาน ความอดทน และความตั้งใจเป็นสิ่งที่จำเป็นต่อการทำโครงการทุกๆ โครงการ ซึ่งจะนำไปสู่การพัฒนาโครงการให้สำเร็จไปได้ด้วยดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ในการเขียนโปรแกรมหรือจะทำโครงการใดๆ ควรจะเขียน Flow Chart ขั้นตอนหรือกระบวนการทำงานเป็นลำดับแรก เพื่อที่จะได้มองเห็นภาพรวมของโครงการและสามารถปรึกษาผู้ที่ปรึกษาหรือเพื่อนร่วมทีมปฏิบัติงานได้อย่างเข้าใจซึ่งกันและกัน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] ไมโครคอนโทรลเลอร์. (ออนไลน์).

แหล่งที่มา : <https://www.thaieasyelec.com/article-wiki/basic-electronics/บทความarduino-คืออะไร-เริ่มต้นใช้งาน-arduino.html>

[2] Microsoft Visual Studio. (ออนไลน์).

แหล่งที่มา : <https://sites.google.com/site/kachapot1150/1-1-microsoft-visual-studio-khux-xari?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F&showPrintDialog=1>

[3] ภาษา C#. (ออนไลน์).

แหล่งที่มา : <https://sites.google.com/site/programming42/phaa-c>

[4] NodeMCU ESP8266. (ออนไลน์).

แหล่งที่มา : <https://www.arduinoall.com/article/30/nodemcu-esp8266-esp8285-arduino-1-esp8266-%E0%B8%84%E0%B8%B7%E0%B8%AD>

[5] TCP VS UDP. (ออนไลน์).

แหล่งที่มา : https://uaychaiblog.com/2016/07/12/tcp_vs_udp/

[6] Flow Chart. (ออนไลน์).

แหล่งที่มา : <https://sites.google.com/site/programming42/kar-kheiyng-phang-ngan-flowchart>

ประวัติผู้เขียน

ชื่อ-นามสกุล	นายศตวรรษ รุ่งอนวงศ์
วัน เดือน ปีเกิด	2 มีนาคม พุทธศักราช 2541
ที่อยู่ปัจจุบัน	111/9 ม.6 ต.บางจาก อ.พระประแดง จ.สมุทรปราการ 10130
เบอร์โทรศัพท์	098-3746311
E-mail	sattawatrb310@hotmail.com
ประวัติการศึกษา	
พุทธศักราช 2548-2553	สำเร็จการศึกษาชั้นประถมศึกษา จากโรงเรียนสารสาสน์ประชาอุทิศพิทยาคาร กรุงเทพมหานคร
พุทธศักราช 2554-2556	สำเร็จการศึกษาระดับชั้นมัธยมศึกษาตอนต้น โรงเรียนวัดราชบพิธ กรุงเทพมหานคร
พุทธศักราช 2557-2559	สำเร็จการศึกษาระดับชั้นมัธยมศึกษาตอนปลาย โรงเรียนวัดราชบพิธ กรุงเทพมหานคร
พุทธศักราช 2560-2563	ศึกษาระดับอุดมศึกษา สาขาวิชาวิศวกรรมแมคคาทรอนิกส์ ภาควิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ประวัติการทำงาน	
พุทธศักราช 2562	สหกิจศึกษา บริษัท เอสซีจี แพคเกจจิ้ง จำกัด (มหาชน) ฝ่าย Engineering Design

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้