รายงานสหกิจศึกษาฉบับสมบูรณ์

แขนกลด้วยมอเตอร์ DC

(DC ROBOT ARM)

ณัฐชนน ศุภอติเรก

ภาควิชาวิศวกรรมการวัดและควบคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

รายงานสหกิจศึกษาฉบับสมบูรณ์

แขนกลด้วยมอเตอร์ DC

(DC ROBOT ARM)

ณัฐชนน ศุภอดิเรก

ภาควิชาวิศวกรรมการวัดและควบคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

ชื่อโครงงาน แขนกลด้วยมอเตอร์ DC

นักศึกษา นายณัฐชนน ศุภอดิเรก

ภาควิชา วิศวกรรมการวัดและควบคุม

อาจารย์นิเทศ ผู้ช่วยศาสตราจารย์ ดร.นพดล มณีรัตน์

ผู้นิเทศงาน นายนีรชัย ศิริสมหมาย

สถานประกอบการ บริษัท เด็นโซ่ (ประเทศไทย) จำกัด (โรงงานบางปะกง)

บทคัดย่อ

โครงงานนี้เป็นการออกแบบและจำลองการสร้างแขนกลโดยใช้ Gear Motor 24Vdc เพื่อลดต้นทุนของการผลิตแขนกล เนื่องจากแขนกลที่ใช้อยู่ตามปกติจะใช้ Motor แบบ Stepper หรือ Servo ซึ่งเมื่อรวมกับราคาของอุปกรณ์เสริมที่ต้องใช้ควบคู่กับมอเตอร์จะทำให้ราคาในการผลิตแขนกลหนึ่งชุดค่อนข้างสูงอีกทั้งยังมีระบบทำงานที่ซับซ้อนและเมื่อแขนกลเสียหายพนักงานในสายการผลิตก็ไม่สามารถตรวจสอบได้ด้วยตนเอง ซึ่งระบบแขนกลใหม่ที่ทดลองสร้างขึ้นมาจะใช้ Gear Motor ควบคู่กับ Magnetic Encoder ในการขยับแขน และใช้ Arduino เป็นระบบควบคุม

คำสำคัญ : แขนกลโดยใช้ Gear Motor, Magnetic Encoder, Arduino

Project Title:	DC Robot Arm
Student Intern:	Mr.Natchanon Suppaadirek
Department:	Instrumentation and Control Engineering
Advisor:	Assistant Professor. Dr. Noppadol Maneerat
Mentor:	Mr.Neerachai Sirisommai
Company:	Denso (Thailand) Co., Ltd.

ABSTRACT

This project is to design and simulate the construction of the mechanical arm by using Gear Motor 24 Vdc to reduce the cost of mechanical arm. The mechanical arm that normally used is made of Stepper or Servo Motor. The mechanical arm cost and the price of the accessories that used with the motor, will make the price of producing one set of Robot Arm quite high. When the mechanical arm is damaged, employees in the production line cannot check it by themselves because of complexity. However the new mechanical arm system will be made by the Gear Motor along with the Magnetic Encoder and Arduino as the control system

Keywords: Mechanical Arm by Using Gear Motor 24 Vdc Machine, Magnetic Encoder, Arduino

กิตติกรรมประกาศ

ในการจัดทำโครงงานในครั้งนี้สามารถสำเร็จลุล่วงด้วยดีได้เนื่องจากบุคลากรหลายๆ ท่านที่ได้ให้ความช่วยเหลือเป็นอย่างดีในตลอดระยะเวลาการทำโครงงาน 6 เดือน ขอขอบคุณบุคลากรแผนก R&D บริษัท เด็นโซ่ (ประเทศไทย) จำกัด (โรงงานบางปะกง) ที่ให้ความรู้ คำแนะนำ สอนทักษะการใช้ชีวิต ทักษะการทำงาน และคอยสนับสนุนวัสดุอุปกรณ์และเครื่องมือต่างๆ สำหรับการทำโครงงาน

ขอขอบคุณบริษัท เด็นโซ่ (ประเทศไทย) จำกัด (โรงงานบางปะกง) ที่ได้เปิดโอกาสให้ข้าพเจ้าได้มาเก็บเกี่ยวประสบการณ์การทำงานในสถานที่จริง

สุดท้ายนี้ขอขอบคุณเพื่อนร่วมงานทุกท่าน ที่คอยเป็นกำลังใจที่ดีตลอดมาจนจบโครงการ ขอขอบคุณไว้ ณ ที่นี้ หากมีข้อผิดพลาดประการใดให้ถือเป็นความบกพร่องของทางคณะผู้จัดทำ และขออภัยมา ณ ที่นี้ด้วย

นายณัฐชนน ศุภอดิเรก

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญรูป

รูปที่ **หน้า**

สารบัญรูป (ต่อ)

สารบัญตาราง

ตารางที่ **หน้า**

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญ

บริษัท เด็นโซ่ (ประเทศไทย) จำกัด เป็นซัพพลายเออร์ชั้นนำ ซึ่งคุณสามารถพบกับเทคโนโลยีและผลิตภัณฑ์ของกลุ่มบริษัทเด็นโซ่ทั่วโลกในอุตสาหกรรมหลายสาขา ตั้งแต่ยานยนต์ไปจนถึงหุ่นยนต์และระบบจัดการพลังงานในบ้าน เทคโนโลยีหรือผลิตภัณฑ์บางรายการอาจไม่มีจำหน่ายในบางประเทศ ดังนั้น บริษัทเด็นโซ่ต้องผลิตอุปกรณ์เป็นจำนวนมากในแต่ละวัน ซึ่งในหลายๆ กระบวนการผลิตจะมีการนำเทคโนโลยีแขนกลมาติดตั้ง

แขนกลเป็นหุ่นยนต์ชนิดหนึ่งที่นำมาใช้งานในวงการอุตสาหกรรมการผลิต ได้ถูกนำมาใช้แทนแรงงานมนุษย์ในงานที่ต้องทำอย่างต่อเนื่อง, งานที่ต้องทำซ้ำๆ, งานที่เป็นอันตราย, งานที่หนักและยากเกินที่มนุษย์จะทำไหว ปกติมนุษย์ก็สามารถทำงานได้ทุกอย่าง แต่ข้อจำกัดของมนุษย์นั้นไม่สามารถทำงานได้อย่างต่อเนื่องยาวนานจะเกิดความเหน็ดเหนื่อยเมื่อยล้าจึงต้องมีการพักผ่อน ส่วนข้อเสียก็มี เช่น มีราคาสูง ต้องมีผู้ชำนาญการในการควบคุมหุ่นยนต์

ดังนั้นจากปัญหาที่กล่าวมาข้างต้น การสร้างแขนกลด้วย Gear Motor VDC จึงเป็นการลดต้นทุนในการผลิตแขนกล ลดความซับซ้อนของระบบปฏิบัติการ ง่ายต่อการซ่อมบำรุง

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อลดต้นทุนในการผลิตแขนกล
2. ลดความซับซ้อนของระบบการทำงาน

1.3 ขอบเขตของโครงงาน

1. ออกแบบวงจรอิเล็กทรอนิกส์และโปรแกรมควบคุมแขนกล
2. สร้างแบบจำลองแขนกลโดยใช้เกียร์มอเตอร์ระบบไฟฟ้ากระแสตรง
3. แขนกลสามารถขยับได้ตามที่ต้องการ
4. วงจรของแขนกลมีความเรียบง่าย

1.4 วิธีดำเนินการ

1. วางแผนการดำเนินงาน
2. ศึกษาการทำงานของ Gear Motor VDC
3. ศึกษามอเตอร์ที่มีความละเอียดสูง เช่น Stepper Motor เพื่อนำมาเปรียบเทียบข้อแตกต่าง
4. ศึกษาอุปกรณ์ที่สามารถนำมาใช้เป็น Feedback ได้ เช่น ตัวต้านทานปรับค่าได้ Encoder
5. ทำการออกแบบแขนกลจำลองสำหรับทดสอบด้วยโปรแกรม Solid Edge
6. ปริ้นชิ้นส่วนแขนกลด้วยเครื่อง Flashforge 3D Printer
7. เขียนโปรแกรมสำหรับควบคุมแขนกล
8. ทดสอบการทำงานและปรับปรุงแก้ไข

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ประโยชน์ต่อตนเอง

การที่ได้มีโอกาสเข้าร่วมในโครงการสหกิจศึกษากับทางสถานประกอบการบริษัท เด็นโซ่ (ประเทศไทย) จำกัด (โรงงานบางปะกง) นั้น ถือว่าได้เป็นการมาเก็บเกี่ยวประสบการณ์ ซึ่งการที่ตนเองนั้นได้เข้ามาทำงานในบริษัทเหมือนกับพนักงานคนหนึ่ง เปรียบเสมือนเป็นการเตรียมตัวก่อนการทำงานจริง และสามารถเพิ่มศักยภาพในการทำงานของตนเองในหลายๆ ด้าน ทั้งด้านการปรับตัว การใช้ความรู้ที่ได้ศึกษาเข้ามาประยุกต์ใช้ในการทำงาน การเรียนรู้สังคมในการทำงาน และยังได้รับข้อคิด ทัศนคติของคนที่ผ่านประสบการณ์การทำงานมานาน

2. ประโยชน์ต่อสถานประกอบการ

การที่ทางสถานประกอบการได้รับนักศึกษาเข้ามาทำโครงการสหกิจศึกษาร่วมกันกับทางสถาบัน แสดงให้เห็นถึงความร่วมมือกันทางวิชาการ ความสัมพันธ์ที่ดีระหว่างสถานประกอบการกับสถาบันการศึกษา และพนักงานประจำมีเวลามากขึ้นที่จะปฏิบัติหน้าที่อื่นที่มีความสำคัญมากกว่า รวมทั้งเป็นวิธีการช่วยคัดเลือกนักศึกษาเข้าเป็นพนักงานประจำในอนาคต โดยไม่จำเป็นต้องมีการทดลองงานก่อน

3. ประโยชน์ต่อสถานศึกษา

ทางสถานศึกษาได้มีการติดต่อหลายหน่วยงานเพื่อส่งนักศึกษาเข้าร่วมโครงการสหกิจศึกษา ดังนั้นการร่วมมือระหว่างสถานศึกษากับสถานประกอบการ จึงถือเป็นการตรวจสอบว่าบุคลากรของทางสถานศึกษามีความรู้ ความสามารถเพียงพอในการทำงานจริงแล้วหรือไม่ ได้ข้อมูลย้อนกลับมาปรับปรุงหลักสูตรการเรียนการสอน

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 Arduino

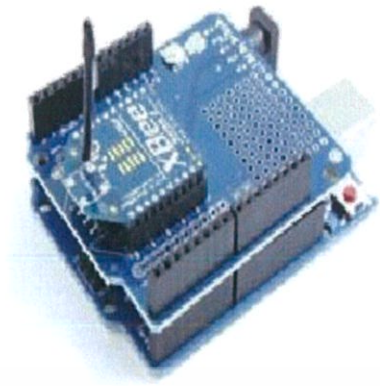
Arduino เป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือ มีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ดหรือโปรแกรมต่อได้อีกด้วย

2.1.1 ส่วนที่เป็นฮาร์ดแวร์

Arduino คือ บอร์ดอิเล็กทรอนิกส์ขนาดเล็ก ที่มีไมโครคอนโทรลเลอร์ (MCU) เป็นชิ้นส่วนหลัก ถูกนำมาประกอบร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่นๆ เพื่อให้ง่ายต่อการใช้งาน โดยบอร์ด Arduino มีหลายรุ่นให้เลือกใช้ โดยในแต่ละรุ่นมีความแตกต่างกันในเรื่องของขนาดของบอร์ดหรือสเปค เช่น จำนวนของขารับส่งสัญญาณ, แรงดันไฟที่ใช้, ประสิทธิภาพของ MCU เป็นต้น

การต่ออุปกรณ์เสริมต่างๆ ของบอร์ด Arduino คือ ผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด ดังรูปที่ 2.1 หรือสามารถเลือกต่อกับบอร์ดเสริม (Arduino Shield) ประเภทต่างๆ ดังรูปที่ 2.2 เช่น Arduino XBee Shield, Arduino Music Shield, Arduino Relay Shield, Arduino Wireless Shield, Arduino GPRS Shield เป็นต้น มาประกอบกับตัวบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้เลย

รูปที่ 2.1 บอร์ด Arduino ต่อกับ LED

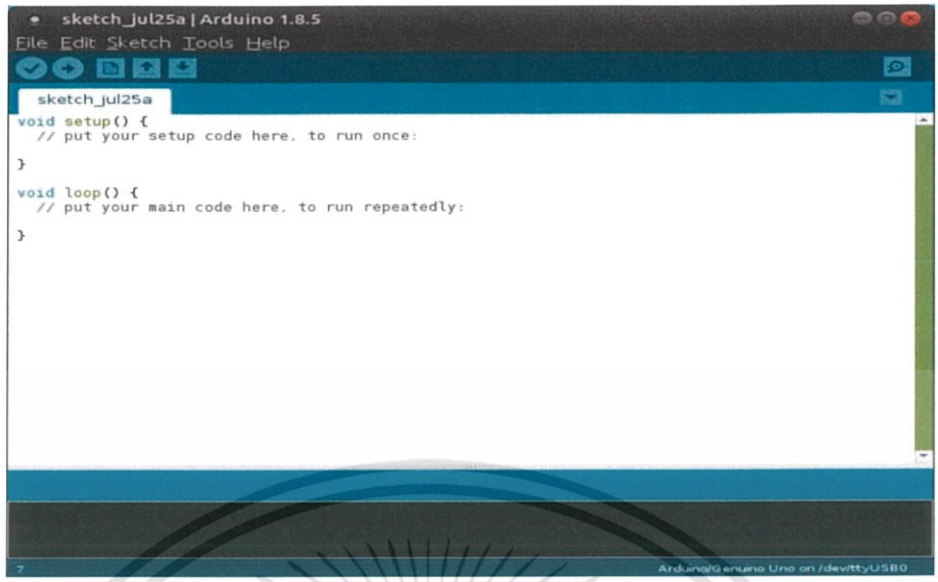


รูปที่ 2.2 บอร์ด Arduino ต่อกับบอร์ด Xbee Shield

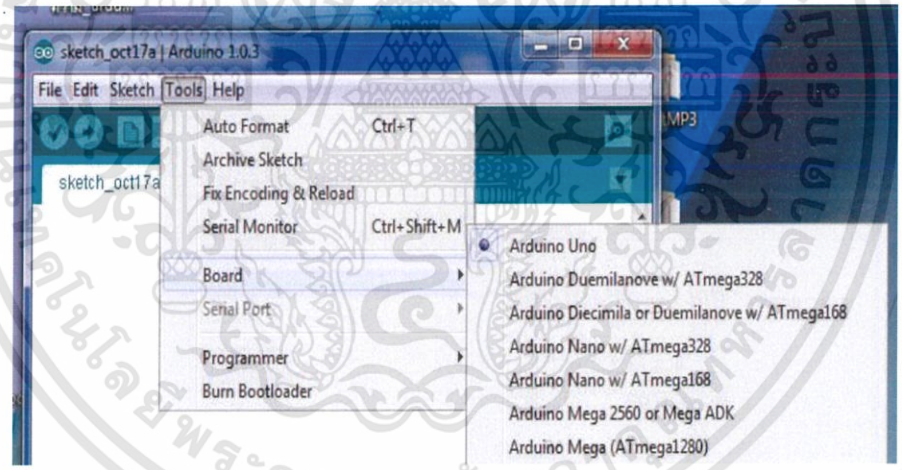
2.1.2 ส่วนที่เป็นซอฟต์แวร์

Arduino จะใช้โปรแกรม ArduinoIDE ในการเขียนโปรแกรมคำสั่งต่างๆ เข้าไปในตัวบอร์ด โดยหน้าตาของตัวโปรแกรมจะเป็นดังรูปที่ 2.3 โดยภาษาที่ใช้ในโปรแกรม Arduino IDE คือ ภาษา Arduino (ภาษา C/C++)

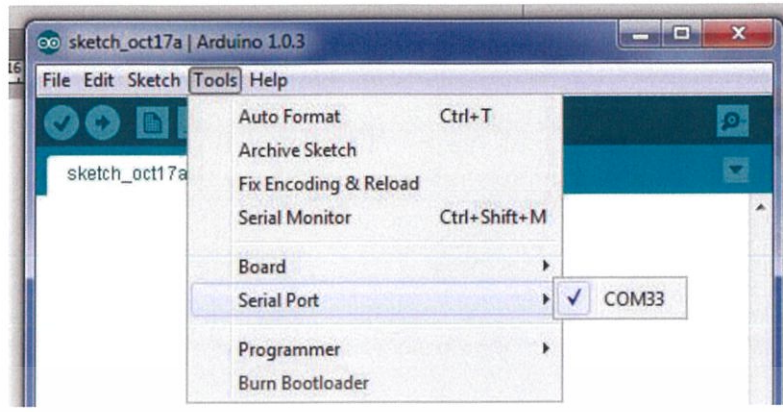
เมื่อผู้ใช้งานต้องการ Upload โปรแกรมเข้าไปในตัวบอร์ด ขั้นตอนแรกต้องทำการเลือกรุ่นบอร์ด Arduino ที่ใช้ดังรูปที่ 2.4 และหมายเลข Comport ดังรูปที่ 2.5 หลังจากนั้นกดปุ่ม Verify เพื่อตรวจสอบความถูกต้องและ Compile โปรแกรมจากนั้นกดปุ่ม Upload โปรแกรมไปยังบอร์ด Arduino ผ่านทางสาย USB เมื่อ Upload เรียบร้อยแล้ว จะแสดงข้อความแถบข้างล่าง “Done Uploading” และบอร์ดจะเริ่มทำงานตามที่เขียนโปรแกรมไว้ได้ทันที ดังรูปที่ 2.6



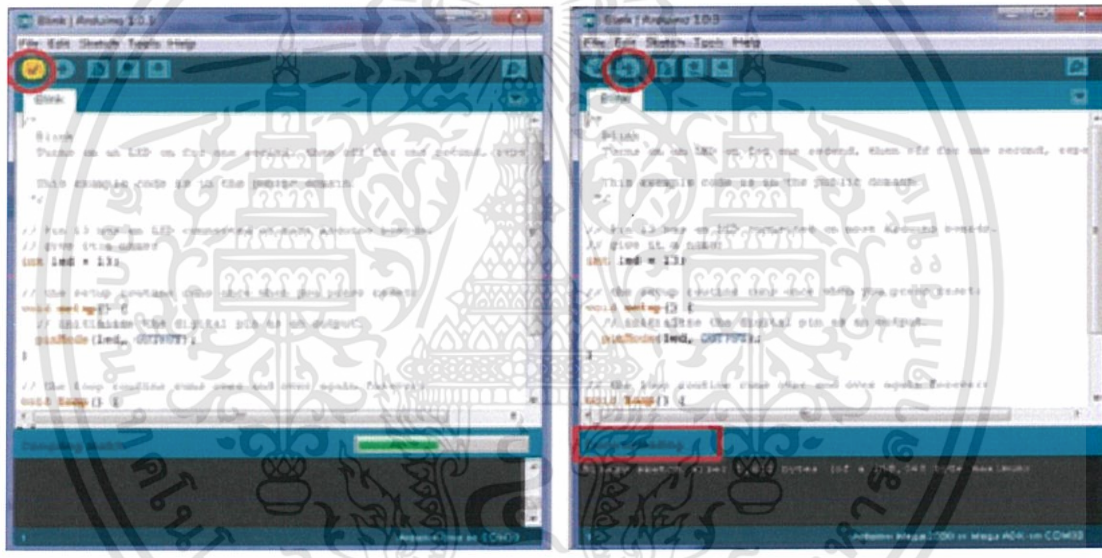
รูปที่ 2.3 หน้าจอโปรแกรม ArduinoIDE



รูปที่ 2.4 เลือกรุ่นบอร์ด Arduino ที่ต้องการ Upload



รูปที่ 2.5 เลือกหมายเลข Comport ของบอร์ด



รูปที่ 2.6 กดปุ่ม Verify และ Compile เมื่อเสร็จสิ้นจะแสดงข้อความแถบข้างล่าง "Done Uploading"

2.1.3 Layout ของ Arduino

ส่วนประกอบพื้นฐานบนบอร์ด Arduino ดังรูปที่ 2.7 ซึ่งถ้าเป็นบอร์ดที่มีขนาดใหญ่จำนวน Pin ก็จะมากขึ้นตามไปด้วย

1. USB Port ใช้สำหรับต่อกับ Computer เพื่ออัพโหลดโปรแกรมเข้า MCU และจ่ายไฟให้กับบอร์ด

2. Reset Button เป็นปุ่ม Reset ใช้กดเมื่อต้องการให้ MCU เริ่มการทำงานใหม่

3. ICSP Port ของ Atmega16U2 เป็นพอร์ตที่ใช้โปรแกรม Visual Comport บน Atmega16U2

4. I/O Port เป็น Digital I/O ตั้งแต่ขา D0 ถึง D13 นอกจากนี้บาง Pin จะทำหน้าที่อื่นๆ เพิ่มเติมด้วยเช่น Pin0,1 เป็นขา Tx, Rx Serial, Pin 3, 5, 6, 9, 10 และ 11 เป็นขา PWM

5. ICSP Port Atmega328 เป็นพอร์ตที่ใช้โปรแกรม Bootloader

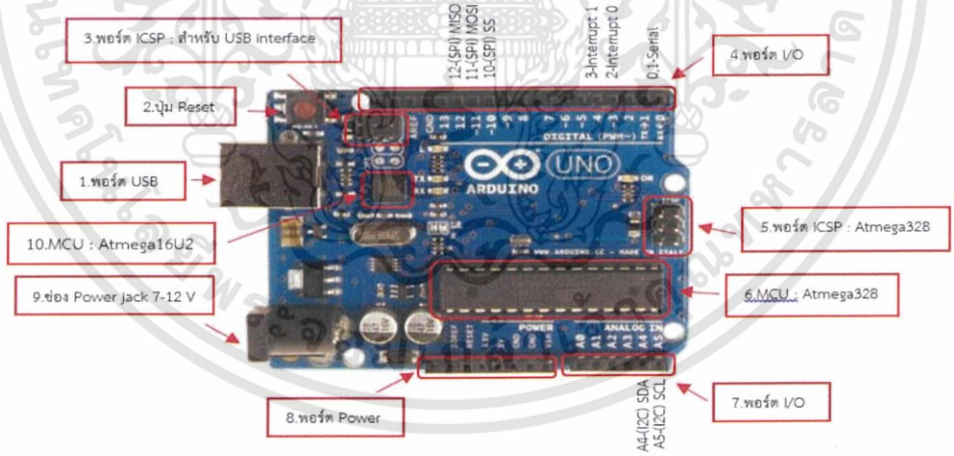
6. MCU Atmega328 เป็น MCU ที่ใช้บนบอร์ด Arduino

7. I/O Port นอกจากจะเป็น Digital I/O แล้ว ยังเปลี่ยนเป็นช่องรับสัญญาณอนาล็อก ตั้งแต่ขา A0-A5

8. Power Port: ไฟเลี้ยงของบอร์ดเมื่อต้องการจ่ายไฟให้กับวงจรภายนอก ประกอบด้วยขาไฟเลี้ยง +3.3 โวลต์, +5 โวลต์, GND, V_{in}

9. Power Jack: รับไฟจาก Adapter โดยที่แรงดันอยู่ระหว่าง 7-12 โวลต์

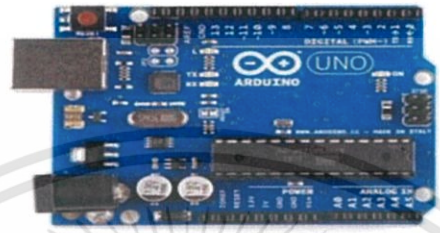
10. MCU ของ Atmega16U2 เป็น MCU ที่ทำหน้าที่เป็น USB to Serial โดย Atmega328 จะติดต่อกับ Computer ผ่าน Atmega16U2



รูปที่ 2.7 Layout ของ Arduino Uno R3

2.1.4 Arduino รุ่นต่างๆ

1. Arduino Uno R3 เป็นบอร์ด Arduino ที่ได้รับความนิยมมากที่สุดเนื่องจากราคาไม่แพง ส่วนใหญ่โปรเจคและ Library ต่างๆ ที่พัฒนาขึ้นมา Support จะอ้างอิงกับบอร์ดนี้เป็นหลักและข้อดีอีกอย่างคือ กรณีที่ MCU เสีย ผู้ใช้งานสามารถซื้อมาเปลี่ยนเองได้ง่าย ดังรูปที่ 2.8

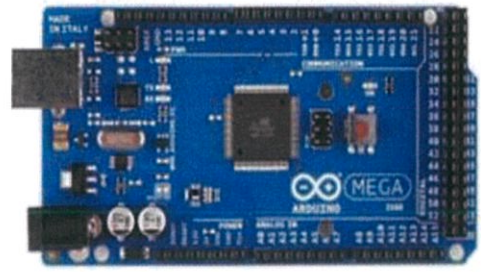


รูปที่ 2.8 Arduino Uno R3

2. Arduino Uno SMD เป็นบอร์ดที่มีคุณสมบัติและการทำงานเหมือนกับบอร์ด Arduino UNO R3 ทุกประการ แต่จะแตกต่างกันที่ Package ของ MCU ซึ่งบอร์ดนี้จะมี MCU ที่เป็น Package SMD (Arduino UNO R3 มี MCU ที่เป็น Package DIP) ดังรูปที่ 2.9

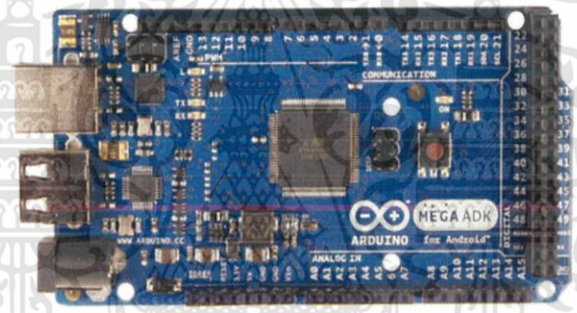
รูปที่ 2.9 Arduino Uno SMD

3. Arduino Mega 2560 R3 เป็นบอร์ด Arduino ที่ออกแบบมาสำหรับงานที่ต้องใช้ I/O มากกว่า Arduino Uno R3 เช่น งานที่ต้องการรับสัญญาณจาก Sensor หรือควบคุมมอเตอร์ Servo หลายๆ ตัว ทำให้ Pin I/O ของบอร์ด Arduino Uno R3 ไม่สามารถรองรับได้ทั้งนี้บอร์ด Mega 2560 R3 ยังมีความหน่วยความจำแบบ Flash มากกว่า Arduino Uno R3 ทำให้สามารถเขียนโปรแกรมเข้าไปได้มากกว่าในความเร็วของ MCU ที่เท่ากัน ดังรูปที่ 2.10



รูปที่ 2.10 Arduino Mega 2560 R3

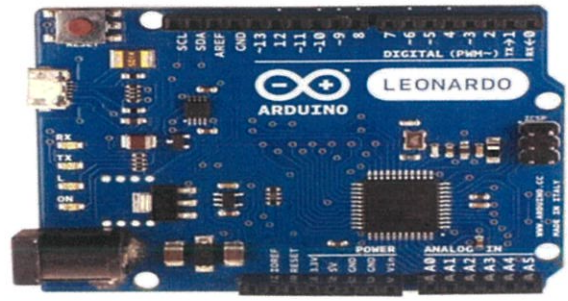
4. Arduino Mega ADK เป็นบอร์ดที่ออกแบบมาให้บอร์ด Mega 2560 R3 สามารถติดต่อกับอุปกรณ์ Android Device ผ่านพอร์ต USB Host ของบอร์ดได้ ดังรูปที่ 2.11



รูปที่ 2.11 Arduino Mega ADK

5. Arduino Leonardo การทำงานจะคล้ายกับบอร์ด Arduino Uno R3 แต่มีการเปลี่ยน MCU ตัวใหม่เป็น ATmega32U4 ซึ่งมีโมดูลพอร์ต USB มาด้วยบนชิป (แตกต่างจากบอร์ด Arduino UNO R3 หรือ Arduino Mega 2560 ที่ต้องใช้ชิป ATmega16U2 ร่วมกับ Atmega328 ในการเชื่อมต่อกับพอร์ต USB) ดังรูปที่ 2.12

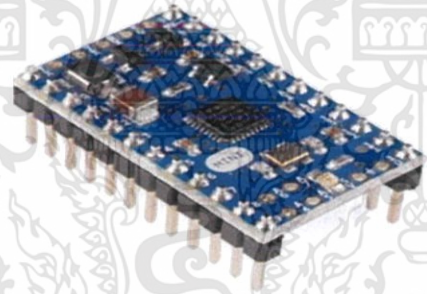
ข้อควรระวัง : เนื่องจาก MCU เป็นคนละเบอร์กับ Arduino Uno R3 อาจจะทำให้บอร์ด Shield บางตัวหรือ Library ใช้ร่วมกันกับบอร์ด Arduino Leonardo ไม่ได้ ผู้ใช้งานจำเป็นต้องตรวจสอบก่อนใช้งาน



รูปที่ 2.12 Arduino Leonardo

6. Arduino Mini 05 เป็นบอร์ด Arduino ขนาดเล็กที่ใช้ MCU เบอร์ ATmega328 เบอร์เดียวกับบอร์ด Arduino UNO R3 ดังรูปที่ 2.13

ข้อแตกต่าง : บอร์ด Arduino Mini 05 จะไม่มีพอร์ต USB มาให้ ผู้ใช้งานต้องต่อกับบอร์ด USB to Serial Converter เพิ่มเมื่อต้องการโปรแกรมบอร์ด

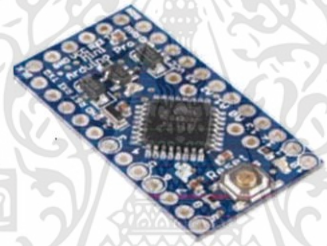


รูปที่ 2.13 Arduino Mini 05

7. Arduino Pro Mini 328 3.3 โวลต์ เป็นบอร์ด Arduino ขนาดเล็กที่ใช้ MCU เบอร์ ATmega328 ซึ่งจะคล้ายกับบอร์ด Arduino Mini 05 แต่บนบอร์ดจะมี Regulator 3.3 โวลต์ ชุดเดียวเท่านั้น ระดับแรงดันไฟที่ขา I/O คือ 3.3 โวลต์ ดังรูปที่ 2.14

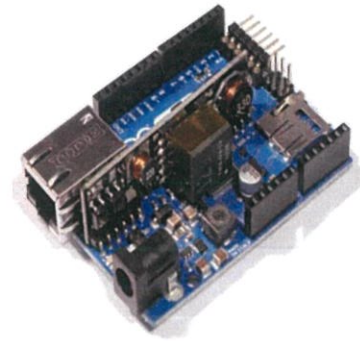
รูปที่ 2.14 Arduino Pro Mini 328 3.3 โวลต์

8. Arduino Pro Mini 328 5 โวลต์ เป็นบอร์ด Arduino ขนาดเล็ก ที่ใช้ MCU เบอร์ ATmega328 เช่นเดียวกับบอร์ด Arduino Mini 05 แต่บนบอร์ดจะมี Regulator 5 โวลต์ ชุดเดียว เท่านั้น ระดับแรงดันไฟที่ขา I/O คือ 5 โวลต์ ดังรูปที่ 2.15



รูปที่ 2.15 Arduino Pro Mini 328 5 โวลต์

9. Arduino Ethernet with PoE Module เป็นบอร์ด Arduino ที่ใช้ MCU เบอร์เดียวกับ Arduino Uno SMD ในบอร์ดมีชิป Ethernet และช่องสำหรับเสียบ SD Card รวมทั้งโมดูล POE ทำให้บอร์ดนี้สามารถใช้แหล่งจ่ายไฟจากสาย LAN ได้โดยตรง โดยไม่ต้องต่อ Adapter เพิ่มแต่บอร์ด Arduino Ethernet with PoE Module นี้จะไม่มีพอร์ต USB ทำให้เวลาโปรแกรมต้องต่อบอร์ด USB to Serial Converter เพิ่มเติม ดังรูปที่ 2.16

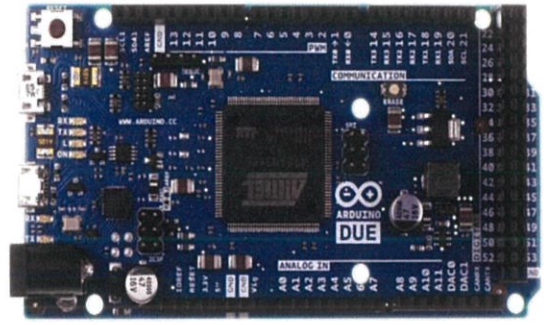


รูปที่ 2.16 Arduino Ethernet with PoE Module

10. Arduino Ethernet Without PoE Module บอร์ดนี้จะตัดโมดูล POE ออกไป ต้องใช้ไฟจากพอร์ต Power Jack เท่านั้น คุณสมบัติอื่นๆ จะเหมือนกับบอร์ด Arduino Ethernet with PoE Module ดังรูปที่ 2.17

รูปที่ 2.17 Arduino Ethernet Without PoE Module

11. Arduino Due เป็นบอร์ด Arduino ที่เปลี่ยนชิป MCU ใหม่ซึ่งจากเดิมเป็นตระกูล AVR เปลี่ยนเป็นเบอร์ AT91SAM3X8E (ตระกูล ARM Cortex-M3) แทน ทำให้การประมวลผลเร็วขึ้นแต่ยังคงรูปแบบโค้ดโปรแกรมของ Arduino ที่ง่ายอยู่ ดังรูปที่ 2.18



รูปที่ 2.18 Arduino Due

เมื่อนำ Arduino ทั้งหมดที่กล่าวถึงข้างต้นมาเปรียบเทียบกันจะได้ข้อสรุปตาม ตารางที่ 2.1 ดังนี้

ตารางที่ 2.1 เปรียบเทียบคุณสมบัติของ Arduino

	Processor					Input / Output						
	Family	SRAM	FLASH	EEPROM	Clock	Digital I/O	Analog In	ADC Bits	PWM	UART	Analog Out	DAC Bits
Arduino UNO R3	ATmega328	2k	32k	1k	16MHz	14	6	10	6	1	N/A	N/A
Arduino UNO SMD	ATmega328	2k	32k	1k	16MHz	14	6	10	6	1	N/A	N/A
Arduino Mega 2560 R3	ATmega2560	8k	256k	4k	16MHz	54	16	10	14	4	N/A	N/A
Arduino Mega ADK	ATmega2560	8k	256k	4k	16MHz	54	16	10	14	4	N/A	N/A
Arduino Leonardo	ATmega32U4	2.5k	32k	1k	16MHz	25	12	10	7	1	N/A	N/A
Arduino Mini 05	ATmega328	2k	32k	1k	16MHz	14	6	10	6	1	N/A	N/A
Arduino Pro Mini 328 - 3.3V	ATmega328	2k	32k	1k	8MHz	14	6	10	6	1	N/A	N/A
Arduino Pro Mini 328 – 5V	ATmega328	2k	32k	1k	16MHz	14	6	10	6	1	N/A	N/A
Arduino Ethernet with PoE module	ATmega328	2k	32k	1k	16MHz	9	6	10	4	1	N/A	N/A
Arduino Ethernet without PoE module	ATmega328	2k	32k	1k	16MHz	9	6	10	4	1	N/A	N/A
Arduino DUE	SAM3X8E	96kb	512k	N/A	84MHz	70	12	12	12	4	2	12

ตารางที่ 2.1 เปรียบเทียบคุณสมบัติของ Arduino (ต่อ)

	Processor					Power				Connectivity				
	Family	SRAM	FLASH	EEPROM	Clock	VCC	Vin Range	5V	3V3	USB-Serial	I2C	Ethernet	USB-Host	SD Card
Arduino UNO R3	ATmega328	2k	32k	1k	16MHz	5V	7-12V	Yes	Yes	ATmega16U2	1	No	No	No
Arduino UNO SMD	ATmega328	2k	32k	1k	16MHz	5V	7-12V	Yes	Yes	ATmega16U2	1	No	No	No
Arduino Mega 2560 R3	ATmega2560	8k	256k	4k	16MHz	5V	7-18V	Yes	Yes	ATmega16U2	1	No	No	No
Arduino Mega ADK	ATmega2560	8k	256k	4k	16MHz	5V	7-18V	Yes	Yes	ATmega16U2	1	MAX3421E	No	No
Arduino Leonardo	ATmega32U4	2.5k	32k	1k	16MHz	5V	7-12V	Yes	Yes	Built-In	1	No	No	No
Arduino Mini 05	ATmega328	2k	32k	1k	16MHz	5V	7V-9V	Yes	No	N/A	1	No	No	No
Arduino Pro Mini 328 - 3.3V	ATmega328	2k	32k	1k	8MHz	3.3V	5V-12V	No	Yes	N/A	1	No	No	No
Arduino Pro Mini 328 – 5V	ATmega328	2k	32k	1k	16MHz	5V	7V-12V	Yes	No	N/A	1	No	No	No
Arduino Ethernet with PoE module	ATmega328	2k	32k	1k	16MHz	5V	6-18V	Yes	Yes	N/A	1	No	No	No
Arduino Ethernet without PoE module	ATmega328	2k	32k	1k	16MHz	5V	6-18V	Yes	Yes	N/A	1	No	No	No
Arduino DUE	SAM3X8E	96kb	512k	N/A	84MHz	3.3V	7-12V	No	VCC	Built-In	2	No	Yes	No

2.2 มอเตอร์ไฟฟ้า

มอเตอร์ไฟฟ้ามีอยู่ทุกที่พบเห็นได้ทั่วไป ตามบ้าน โรงเรียน โรงพยาบาล โรงงานอุตสาหกรรม เมื่อผู้ใช้งานต้องการเลือกใช้มอเตอร์ไฟฟ้าก็จำเป็นต้องเรียนรู้พื้นฐานของมอเตอร์ไฟฟ้าและวิธีการเลือกซื้อที่ถูกต้องเสียก่อน เนื่องจากมีผู้ขายมอเตอร์ไฟฟ้านั้นจะสอบถามว่าคุณต้องการมอเตอร์แบบใด คำถามแรกที่ผู้ใช้จะเจอคือ มอเตอร์แบบไหนเหมาะสมกับการใช้งานของคุณและมีข้อกำหนดอะไรบ้าง

มอเตอร์ไฟฟ้าทำงานโดยการแปลงพลังงานไฟฟ้าให้เป็นพลังงานกลเพื่อให้เกิดการหมุน พลังงานจะถูกสร้างขึ้นภายในมอเตอร์ผ่านปฏิกิริยาระหว่างสนามแม่เหล็กของกระแสไฟฟ้าสลับ (AC) หรือไฟฟ้ากระแสตรง (DC) แรงบิดขึ้นอยู่กับกระแสที่เพิ่มขึ้นตามกฎของโอห์มคือ

$$V = I*R \tag{2.1}$$

โดยที่แรงดันไฟฟ้า (Voltage) ต้องเพิ่มขึ้นเมื่อความต้านทานเพิ่มขึ้นในขณะที่กระแสไฟฟ้ายังคงเท่าเดิม มอเตอร์ไฟฟ้ามีการใช้งานในอุตสาหกรรมทั่วไป เช่น เครื่องเป่าลม เครื่องจักรขนาดใหญ่ พัดลม ปั๊ม งานขนาดเล็กที่ต้องใช้การเคลื่อนไหว เช่น หุ่นยนต์ (Robotics) หรือโมดูลที่มีล้อแบบต่างๆ

2.2.1 กฎ 3 ข้อในการเลือกมอเตอร์ไฟฟ้า

1. กระแสไฟฟ้า (Voltage) เป็นสิ่งที่ทำให้มอเตอร์ทำงานได้และกระแสไฟฟ้าที่มากเกินไปจะเกิดความเสียหายกับมอเตอร์ สำหรับมอเตอร์กระแสตรงการใช้งานและมีความสำคัญ ดังนั้นจึงต้องรู้ก่อนว่า มอเตอร์ที่จะนำไปใช้ต้องใช้กับกระแสไฟประเภทไหน เช่น ไฟ 220 โวลต์ (Single Phase) หรือไฟ 380 โวลต์ (Three Phase) กระแสไฟฟ้าที่ใช้งานอยู่คือ ค่าเฉลี่ยของกระแสที่มอเตอร์คาดว่าจะอยู่ภายใต้แรงบิดทั่วไป กระแสไฟฟ้าสถิตใช้แรงบิดเพียงพอสำหรับมอเตอร์เพื่อให้ทำงานที่การหยุดกลางคัน (0 RPM) ต้องควบคุมไม่ให้กระแสไฟฟ้าเกินและควรมีแผงระบายความร้อนเพื่อป้องกันไม่ให้ขดลวดไหม้

2. แรงดันไฟฟ้า (Current) ใช้เพื่อให้กระแสไฟฟ้าไหลไปในทิศทางเดียวกันและเพื่อป้องกันกระแสย้อนกลับ แรงดันไฟฟ้าที่สูงขึ้น แรงบิดที่สูงขึ้น แรงดันไฟฟ้า จะบอกประสิทธิภาพของมอเตอร์ระหว่างทำงาน ต้องแน่ใจว่าใช้ไฟกี่โวลต์ หากใช้น้อยไปมอเตอร์ไม่หมุน หากแรงดันไฟมากไปมอเตอร์

อาจไหม้ได้การทำงานของมอเตอร์ต้องคำนึงถึงแรงบิดด้วย เพราะงานบางอย่างจำเป็นต้องอาศัยแรงบิดที่เพียงพอ แรงบิดมีความสำคัญมากกว่าความเร็ว

3. ความเร่งหรือความเร็ว (RPM) โดยทั่วไปมอเตอร์ทำงานได้อย่างมีประสิทธิภาพด้วยความเร็วสูงสุด แต่หากต้องใช้ระบบเกียร์ การเพิ่มเกียร์จะลดประสิทธิภาพของมอเตอร์ดังนั้นโปรดคำนึงถึงความเร็วและแรงบิดที่ลดลงเช่นกัน

โดยแรงบิดของมอเตอร์ไฟฟ้าสามารถหาได้จากสูตร

$$P = \left(\frac{2*(22/7*T*N)}{60}\right) \quad (2.2)$$

P = กำลังงาน หน่วยเป็น วัตต์ (1HP = 745.7 วัตต์)

T = แรงบิด หน่วยเป็น N-m.

N = ความเร็วรอบของมอเตอร์ไฟฟ้า หน่วยเป็น RPM. (round/minute)

2.2.2 ประเภทของมอเตอร์ไฟฟ้า

2.2.2.1 ดีซีมอเตอร์ (DC Motor) ส่วนใหญ่เป็นมอเตอร์ขนาดเล็กมีอยู่หลายประเภทแต่แบ่งเป็น 5 ชนิดหลักๆ คือ

1. มอเตอร์แบบมีแปรงถ่าน (Brushed) พบได้ในเครื่องใช้ไฟฟ้าจำนวนมาก เช่น พวกของเล่น รถยนต์ โดยใช้การควบคุมทิศทางการหมุนของมอเตอร์ เช่น บังคับของเล่นให้สามารถเดินหน้าถอยหลังได้ มอเตอร์ชนิดนี้ราคาไม่แพง กระบวนการผลิตง่ายไม่ซับซ้อน มีแรงบิดดีมากที่ความเร็วรอบต่ำ (วัดความเร็วรอบต่อนาทีหรือ RPM) ข้อเสียคือ ต้องเปลี่ยนแปรงถ่านเนื่องจากเกิดความสึกหรอ เพราะความร้อนและมีเสียงรบกวนจากสนามแม่เหล็กไฟฟ้าได้

2. มอเตอร์แบบไม่มีแปรงถ่าน (Brush Less) ใช้แม่เหล็กถาวรในชุดโรเตอร์ นิยมในกลุ่มคนชอบงานอดิเรก สำหรับเครื่องบินและการประยุกต์ใช้ในยานพาหนะภาคพื้นดิน มอเตอร์ชนิดนี้มีประสิทธิภาพสูง การบำรุงรักษาน้อย เสียงรบกวนต่ำ และให้กำลังงานสูงกว่ามอเตอร์แบบมีแปรงถ่าน นอกจากนี้ยังสามารถผลิตได้เป็นจำนวนมากและคล้ายมอเตอร์เอซี (AC Motor) ที่มีความเร็วรอบคงที่ยกเว้นใช้กระแสไฟฟ้าไฟดีซี แต่ก็ยังมีข้อเสียอยู่บ้างคือ การควบคุมความเร็วค่อนข้างยากและ

ต้องมีโหลดเริ่มต้นต่ำและอาจจะต้องมีเกียร์บ็อกซ์ (Gearboxes) ซึ่งทำให้มีค่าใช้จ่ายเพิ่มขึ้นและข้อจำกัดด้านสภาพแวดล้อม

3. มอเตอร์แบบสั่น (Vibration Motor) ใช้สำหรับทำให้เกิดการสั่นสะเทือน เช่น โทรศัพท์มือถือ หรือคอลโทรลเลอร์เกม ถูกสร้างขึ้นโดยมอเตอร์ไฟฟ้าที่ไม่สมดุลซึ่งเป็นสาเหตุให้เกิดการสั่นสะเทือน นอกจากนั้นยังสามารถใช้ใน Buzzers เพื่อวัตถุประสงค์ทำให้เกิดเสียงหรือเสียงเตือนภัยรวมถึงกริ่งประตูด้วย

4. สเตปปิ้งมอเตอร์ (Stepping Motor) เป็นมอเตอร์ที่มีความแม่นยำสูง พบในเครื่องพิมพ์ เครื่องมือต่างๆ และระบบที่ต้องการควบคุมกระบวนการผลิตที่ต้องการความแม่นยำและมีแรงบิดที่สูง การควบคุมมอเตอร์ชนิดนี้จะใช้สัญญาณพัลซ์ (Pulse) ในการควบคุมเพื่อให้ตัวขับ (Driver) ส่งแรงดันไฟฟ้าตามสัดส่วนไปยังมอเตอร์ ซึ่งการควบคุมค่อนข้างง่ายแต่จะใช้กระแสสูงสามารถใช้งานได้ดีกับโหลดสูงๆ

5. เซอร์โวมอเตอร์ (Servo Motor) เป็นมอเตอร์ที่ได้รับความนิยมในตลาดและใช้สำหรับการควบคุมตำแหน่งแต่ไม่ต้องการความแม่นยำสูง ใช้ในการควบคุมระยะไกล เช่น รถของเล่น RC และหุ่นยนต์ ประกอบด้วย มอเตอร์ โพเทนชิออมิเตอร์และวงจรควบคุมผ่าน Pulse Width Modulation (PWM) ผ่านการส่งพัลส์ไฟฟ้าไปยังสายควบคุม เซอร์โวสามารถเป็นได้ทั้ง AC Servo หรือ DC Servo สามารถรองรับกระแสไฟฟ้าที่สูงขึ้นได้และใช้สำหรับเครื่องจักรอุตสาหกรรมในขณะที่ DC Servo สำหรับงานอดิเรกขนาดเล็ก

2.2.2.2 เอซีมอเตอร์ (AC Motor) มีด้วยกัน 4 ประเภทหลัก คือ

1. มอเตอร์เหนี่ยวนำ (Induction Motor) เรียกว่ามอเตอร์แบบ “อะซิงโครนัส” เนื่องจากไม่เคลื่อนที่ด้วยอัตราคงที่ หรือหมุนช้ากว่าความถี่ที่ให้มาความแตกต่างระหว่างความเร็วจริงและความเร็วในการซิงโครนัสเป็นสิ่งจำเป็นเพื่อสร้างแรงบิดที่ทำให้เกิดการหมุนในมอเตอร์เหนี่ยวนำ สนามแม่เหล็กที่ล้อมรอบโรเตอร์ของมอเตอร์ทำให้เกิดการกระแสเหนี่ยวนำ

2. ซิงโครนัสมอเตอร์ (Synchronous Motor) จะหมุนด้วยอัตราคงที่เนื่องจากใช้กระแสไฟฟ้าสลับ (AC)

3. มอเตอร์อุตสาหกรรม (Industrial Motor) ได้รับการออกแบบเพื่อใช้งานกับไฟฟ้าสามเฟสหรืองานที่ต้องใช้กำลังมาก เช่น สายพานลำเลียง เครื่องเป่าลม มอเตอร์ไฟฟ้ากระแสสลับพบได้ในเครื่องใช้ไฟฟ้าภายในบ้านและอุปกรณ์อื่นๆ เช่น นาฬิกาพัดลมและดิสก์ไดรฟ์

4. มอเตอร์กันระเบิด (Explosion Proof Motor) มีแตกต่างจากมอเตอร์ทั่วไปตรงที่กล่องขั้ว และตัวถังของตัวมอเตอร์จะดีกว่ามอเตอร์ทั่วไปเพื่อป้องกันประกายไฟไม่ให้เกิดการระเบิดกับวัตถุไวไฟ เช่น ถ่านหิน น้ำมัน และก๊าซปิโตรเคมี

2.3 Step Down

วงจร Buck Converter หรือ วงจร Step Down เป็นวงจรที่ลดแรงดันไฟฟ้าให้ต่ำลงเพื่อเหมาะสมต่อการใช้งาน ดังรูปที่ 2.19

2.3.1 ส่วนประกอบหลักๆ ของวงจร Step Down ประกอบไปด้วย

2.3.1.1 MOSFET ประกอบด้วย

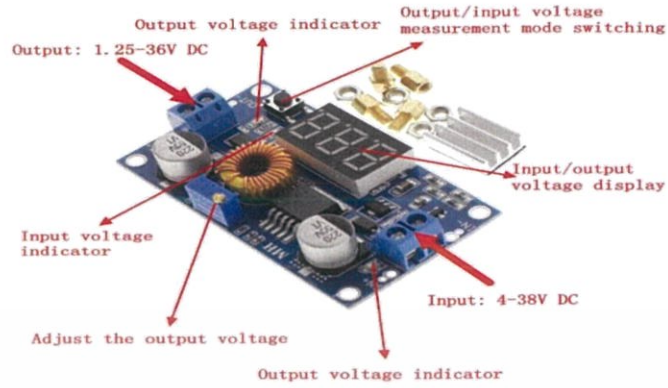
1. GATE เป็นส่วนที่ทำมาจากออกไซด์ของโลหะ โดยสร้างให้เกิดความต่างศักย์ตกคร่อมระหว่างแผ่นสองแผ่นเพื่อสร้างสนามไฟฟ้าเพื่อควบคุมการเข้าออกของสัญญาณไฟฟ้า
2. SOURCE เป็นส่วนขาเข้าของสัญญาณ
3. DRAIN เป็นส่วนขาออกของสัญญาณ

2.3.1.2 Gating Block เป็นสัญญานกระตุ้นให้ MOSFET ทำงาน โดยสัญญาณจะอยู่ในค่า 0 กับ 1

2.3.1.3 Capacitor เป็นอุปกรณ์ไฟฟ้าที่ทำหน้าที่สะสมหรือเก็บประจุไฟฟ้าเปรียบเทียบเหมือนแหล่งจ่ายแรงดันไฟฟ้าหรือแบตเตอรี่ที่ต่ออยู่ในวงจรไฟฟ้า

2.3.1.4 Inductor หลักการทำงานของตัวเหนี่ยวนำใช้หลักการสนามแม่เหล็กตัดผ่านขดลวด จะทำให้เกิดการไหลของกระแสไฟฟ้าในขดลวด ซึ่งจะทำให้เกิดการเหนี่ยวนำขึ้น

2.3.1.5 Diode เป็นชิ้นส่วนอิเล็กทรอนิกส์ชนิดสองขั้วคือ ขั้ว p และขั้ว n ที่ออกแบบและควบคุมทิศทางการไหลของประจุไฟฟ้า มันจะยอมให้กระแสไฟฟ้าไหลในทิศทางเดียว และกั้นการไหลในทิศทางตรงกันข้าม



รูปที่ 2.19 XL4015 DC-DC Converter Step Down

2.4 Encoder

Encoder คือ เซนเซอร์ชนิดหนึ่งที่ทำหน้าที่ในการเข้ารหัสจากระยะทางจากการหมุนรอบตัวเอง และแปลงออกมาเป็นรหัสในรูปแบบของสัญญาณไฟฟ้า โดยสามารถนำเอารหัสเหล่านี้มาแปลงกลับเพื่อหาค่าต่างๆ ที่ต้องการได้ ไม่ว่าจะเป็นระยะทางการหมุน องศาการเคลื่อนที่หรือ ความเร็วรอบแล้วนำมาแสดงผลให้ได้ทราบค่าผ่านหน้าจอแสดงผล เช่น ถ้าต้องการวัดระยะทางจะต้องต่อเข้ากับตัวนับจำนวน เพื่อแสดงผลเป็นระยะทางหรือถ้าต้องการวัดความเร็วรอบ ผู้ใช้งานจะต้องต่อเข้ากับตัววัดพัลส์โดยการประยุกต์ใช้ Encoder นั้นสามารถใช้ทำงานได้อย่างหลากหลาย เช่น กระบวนการประกอบชิ้นส่วนอิเล็กทรอนิกส์ อุตสาหกรรมเซมิคอนดักเตอร์ เครื่องมือวัดต่างๆ เป็นต้น ส่วนการแสดงผลเป็นความเร็วรอบของ RPM, RPS โดยอาศัยสัญญาณที่ผ่านการเข้ารหัสแล้วออกมาเป็นสัญญาณทางไฟฟ้านั้น สามารถแบ่งรูปแบบของการเข้ารหัสได้อีกหลากหลายรูปแบบ เช่น สัญญาณดิจิตอล ศูนย์กับหนึ่งธรรมดาหรือเป็นแบบ Binary Code, BCD Code, Gray Code

ซึ่งในปัจจุบันก็มีสัญญาณที่เป็นการสื่อสารแบบอนุกรมอื่นๆ อีกมากมายที่ถูกนำมาใช้กับ Encoder เช่น SSI, CAN, PROFIBUS, ETHERCAT โดยการสื่อสารแบบต่างๆ เหล่านี้ถูกออกแบบมาเพื่อเพิ่มระยะในการสื่อสารให้ไกลมากขึ้น อีกทั้งยังเป็นการลดปัญหาเรื่องการส่งสัญญาณที่ผิดพลาด และสามารถตอบสนองต่อยุคการผลิตแบบ Industry 4.0 หรือ Smart Factory ได้เป็นอย่างดี

Encoder สามารถแบ่งได้หลายแบบ โดยขึ้นอยู่ว่าจะยึดเกณฑ์ใดเพื่อใช้ในการแบ่ง เช่น สามารถแบ่งได้จากโครงสร้างการใช้งานหรือแบ่งตามโครงสร้างของ Encoder ก็ทำได้ และนอกจากนี้ยังสามารถใช้ประเภทของ Output ของ Encoder มาใช้ในการแบ่งได้อีกเช่นกัน

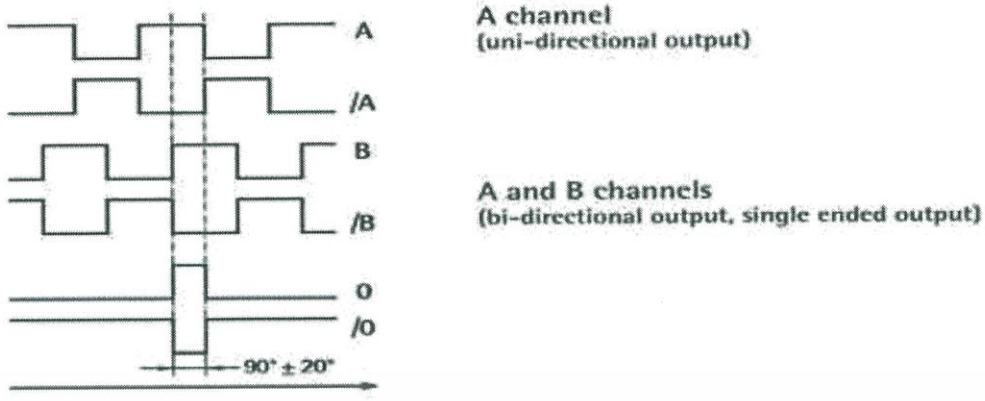
โดยความสำคัญของทั้งหมดที่กล่าวมานั้นจะเป็นส่วนจำเป็นต่อการเลือกใช้ Encoder ให้เหมาะสมกับการใช้งาน แต่ถ้าหากพบเจอกับข้อจำกัดต่างๆ ที่ก่อให้เกิดความลำบากในการทำงานหรือการติดตั้ง สิ่งหนึ่งที่จะสามารถช่วยลดหรือขจัดข้อจำกัดต่างๆ ไปได้นั้นก็คือ อุปกรณ์เสริมของ Encoder ซึ่งอุปกรณ์เหล่านี้จะถูกออกแบบมาเพื่อช่วยงานที่มีข้อจำกัดต่างๆ โดยเฉพาะ

2.4.1 ประเภทของ Encoder

2.4.1.1 แบ่งจากโครงสร้างการใช้งาน สามารถแบ่งได้ 2 ประเภทใหญ่ๆ

1. Incremental Rotary Encoder เป็น Encoder ที่รูปแบบสัญญาณ Output ที่ออกมาเป็นลักษณะของสัญญาณพัลส์ที่เป็นคลื่นรูปสี่เหลี่ยมซึ่งจะไม่เหมือนกับไซน์เวฟ โดยจำนวนพัลส์ที่ออกมานั้นจะมีความสัมพันธ์กับระยะการเคลื่อนที่, ตำแหน่ง, ระยะห่าง, ความเร็ว และความเร่ง นอกจากนี้ยังสามารถระบุได้ถึงทิศทางการหมุนของ Encoder ได้ว่าจะหมุนตามเข็มนาฬิกาหรือทวนเข็มนาฬิกา โดยอาศัยการตรวจจับทิศทางการหมุนจากมุมเฟสของสัญญาณ Output A กับ B ว่าสัญญาณใดเกิดก่อนกันซึ่งจะมีมุมเฟสที่ต่างกันอยู่ 90 องศา ดังรูปที่ 2.20

Encoder แบบนี้ มีข้อเสียตรงที่ไม่สามารถจดจำตำแหน่งแกนหมุนของตัวเองได้ว่าอยู่ที่จุดใด ด้วยเหตุนี้การหมุนกลับไปยังตำแหน่งเริ่มต้น Homing Point นั้นจะทำได้ยากจึงต้องอาศัยการเก็บข้อมูลพัลส์ตั้งแต่เริ่มต้นเพื่อหาค่าตำแหน่ง ซึ่งในบางครั้งอาจจะอ้างอิงจากจุด Zero Point

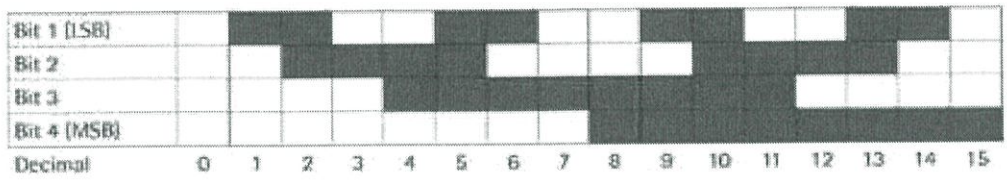


รูปที่ 2.20 ภาพแสดงสัญญาณพัลส์

2. Absolute Rotary Encoder เป็น Encoder ที่ออกแบบมาให้มีรูปแบบสัญญาณ Output ที่เป็นลักษณะของการเข้ารหัส เพื่อต้องการแก้ปัญหาของ Incremental Encoder เนื่องจากสัญญาณ Output ของ Incremental Encoder ไม่สามารถระบุตำแหน่งพัลส์กับตำแหน่งองศาของแกน Encoder ได้ จึงแก้ปัญหาเหล่านี้ได้โดยการใช้ Absolute Rotary Encoder ซึ่งจะใช้รหัสแทนสัญญาณพัลส์ โดยรหัสเหล่านี้ก็จะมีหลากหลายรูปแบบเช่น BCD, Binary, Gray Code ซึ่งจะสามารถแทนค่าตำแหน่งองศาที่แกนของ Encoder หยุดอยู่ได้ดังรูปที่ 2.21 และรูปที่ 2.22

นอกจากนี้ยังเลือกเป็นสัญญาณแบบขนานหรืออนุกรมก็ได้ ซึ่งจะทำให้ได้ตำแหน่งที่ถูกต้องและแม่นยำมากที่สุด ถึงแม้ว่าจะหยุดจ่ายไฟและจ่ายไฟเข้าไปใหม่ก็ยังสามารถบ่งบอกได้ว่าตำแหน่งองศาที่อยู่นั้นคือเท่าใด

รูปที่ 2.21 ภาพแสดงสัญญาณ Output แบบรหัส Binary จำนวน 4 บิต

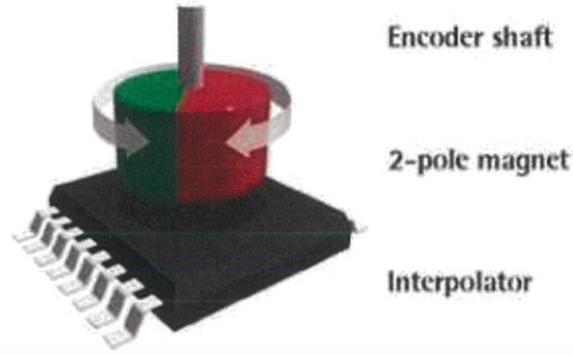


รูปที่ 2.22 ภาพแสดงสัญญาณ Output แบบรหัส Binary จำนวน 4 บิต

2.4.1.2 การแบ่งตามโครงสร้างของ Encoder แบ่งออกได้เป็น 2 ลักษณะได้ดังนี้

1. Rotary Optical Encoder เป็นเซนเซอร์แบบพื้นฐานที่ถูกออกแบบมาใช้งานตั้งแต่เริ่มต้นโดยใช้หลักของ Optical Sensing ร่วมกับ Code Disk และใช้ตัว Scanning Reticle ซึ่งจะเป็นตัวช่วยบังคับแสงให้ไปตรงที่ตัวรับแสง หรือ Phototransistor ได้อย่างแม่นยำ

2. Rotary Magnetic Encoder จากการทำงานของตัว Rotary Optical Encoder ซึ่งได้กล่าวไว้ข้างต้น โครงสร้างภายในนั้นสร้างมาจาก Code Disk ซึ่งเป็นวัสดุใสและมีความเปราะบางไม่ทนต่อการใช้งานที่มีแรงสั่นสะเทือนและการกระแทกได้มาก ดังนั้นจึงได้มีการพัฒนาตัว Rotary Magnetic Encoder โดยจะมีความสามารถในการใช้งานที่สูงกว่าและทนต่อแรงสั่นสะเทือนได้ดีกว่าแบบ Code Disk จึงเป็นการยืดอายุการใช้งานของ Encoder ได้ยาวนานมากยิ่งขึ้น ซึ่งหลักการทำงานของตัว Magnetic Encoder นั้นจะใช้ Maganetic Code Ring ที่เป็นตัวหมุนและใช้เซนเซอร์ที่ใช้ในการตรวจจับ Code จากสนามแม่เหล็ก ดังรูปที่ 2.23 โดยเซนเซอร์ที่ว่านี้มีอยู่ 2 แบบ คือ Magneto-Resistance (MR) และ IC Sensor โดยเซนเซอร์ทั้ง 2 รูปแบบนี้เมื่อมีสนามแม่เหล็กตัดผ่านจะมีการแยกแยะระหว่างขั้วเหนือและขั้วใต้ ซึ่งทำให้เกิดปรากฏการณ์ Hall Effect จากปรากฏการณ์นี้จะทำให้ได้แรงดันไฟฟ้าที่เป็นแบบ Sine Wave ออกมา ต่อจากนั้นจะมีการใช้กระบวนการแปลงสัญญาณจากสัญญาณ Analog to Digital A/D เพื่อนำเอาสัญญาณดิจิตอลเหล่านี้ไปใช้งานต่อไป

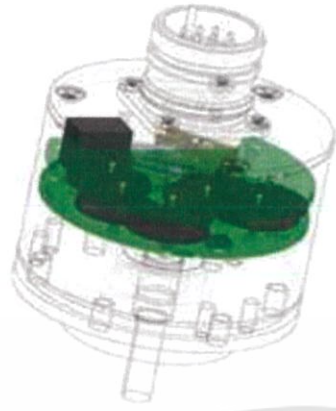


รูปที่ 2.23 โครงสร้างของ Magnetic Code Ring และ IC Sensor ของ Rotary Magnetic Encoder

2.4.1.3 การแบ่งแบบ Special Rotary Encoder นอกจากทั้ง 2 รูปแบบการแบ่งที่ผ่านมาแล้วนั้น ยังมีประเภทพิเศษที่ถูกออกแบบมาเพื่อใช้ในงานที่มีลักษณะเฉพาะ โดยมีรายละเอียดดังนี้

1. Programmable Pulse Rotary Encoder เป็น Encoder ชนิดพิเศษที่สามารถตั้งค่าของจำนวนพัลส์ต่อรอบหรือ Pulse Per Revolution ได้ ซึ่งในปัจจุบันมียี่ห้อ LIKA รุ่น IQ58, CKQ58 ที่สามารถป้อนโปรแกรมเพื่อกำหนดจำนวนพัลส์ที่ต้องการใช้งานได้ ซึ่งทำให้ง่ายต่อการใช้งานและการทำรายการอะไหล่

2. Multi-turn Rotary Encoder เป็น Absolute Rotary Encoder ชนิดหนึ่งที่ถูกออกแบบมาเพื่อแก้ปัญหา Single-turn Absolute Rotary Encoder ที่ไม่สามารถจดจำในส่วนของจำนวนรอบที่หมุนได้ เนื่องจากตัวรหัสออกแบบมาให้ไม่ซ้ำกันได้แค่หนึ่งรอบเท่านั้น ทำให้มีขีดจำกัดในเรื่องของความละเอียด แต่สำหรับ Multi-turn Rotary Encoder จะมีการเพิ่มวงจรนับจำนวนไว้ภายในตัว ซึ่งจะทำให้สามารถรู้ได้ว่าตัว Encoder หมุนไปกี่รอบแล้ว นอกจากนี้ยังเป็นการเพิ่มความละเอียดในการใช้งานได้ด้วย ดังรูปที่ 2.24



รูปที่ 2.24 ภาพแสดงโครงสร้างของ Multi-turn Rotary Encoder

2.4.2 ประเภท Output ของ Encoder

เมื่อ Encoder ทำงานจะมีการสร้างสัญญาณทางไฟฟ้าออกมาหลังจากการหมุนของแกน สัญญาณจากนั้นจะถูกนำไปประมวลผล ซึ่งในอดีตสัญญาณที่ได้จากตัวเซนเซอร์ประเภทนี้นั้นจะได้เป็นแบบสัญญาณอนาล็อก ซึ่งจะมีลักษณะเป็นคลื่นรูปไซน์และจะแปรเปลี่ยนระดับของแรงดันตามความเร็วรอบ ได้ถูกจำกัดในเรื่องของการใช้งานไว้ที่การวัดความเร็วรอบ เนื่องจากความละเอียดค่อนข้างต่ำ แต่สำหรับปัจจุบันนั้น Encoder จะมีสัญญาณ Output ที่เป็นแบบดิจิตอลเกือบทั้งหมด ดังนั้นการเลือกใช้งานตัว Encoder เพื่อให้มีประสิทธิภาพสูงสุด ผู้ใช้งานจำเป็นต้องศึกษาลักษณะสัญญาณ Output แบบต่างๆ และรหัสของ Output รวมถึงสัญญาณสื่อสารแบบต่างๆ เพื่อที่จะทำให้การใช้งานสามารถทำงานได้ถูกต้อง โดยสามารถอธิบายลักษณะของสัญญาณ Output ได้เป็นส่วนต่างๆ ดังนี้

2.4.2.1 Output Code (Parallel) เป็นการรหัสตำแหน่งของ Encoder ด้วยรหัสดิจิตอล เพื่อให้สามารถนำไปต่อใช้งานกับระบบประมวลผลแบบดิจิตอลได้ โดยมีการกำหนดให้สายสัญญาณแต่ละเส้นแทนค่าบิตของข้อมูลที่ส่งออกมา โดยรหัสดิจิตอลที่นิยมนำมาใช้ในตัว Encoder มีดังนี้

1. Binary Code เป็นสัญญาณ Code Output ที่อยู่ในรูปของเลขฐาน 2 โดยจะใช้เลข “0”, “1” ในการแสดงสถานะต่างๆ โดยเลขฐาน 2 นั้นมีประสิทธิภาพในการเข้ารหัสเลขฐาน 10 ที่ใช้งานอยู่ แต่ก็ยังมีปัญหาเกิดขึ้นบ้างในเรื่องของการหน่วงของสัญญาณที่เกิดขึ้นในแต่ละบิต ซึ่งอาจทำให้เกิดการเข้ารหัสผิดพลาด และทำให้ค่าที่อ่านออกมาได้ ไม่ตรงตามค่าที่ตำแหน่งของ Encoder อยู่จริง

2. Gray Code เป็นสัญญาณ Code Output ที่ออกแบบโดย Frank Gray ในปี ค.ศ. 1947 ซึ่งประสบความสำเร็จเป็นอย่างมากในการแก้ปัญหาการปลอมบิต Spurious Output จากการถูกรบกวนทางไฟฟ้า ซึ่งจากรูปที่ 2.25 จะเห็นได้ว่าในการเปลี่ยนค่าจาก 1 เป็น 2 ในเลขฐาน 10 นั้น ในรหัส Binary Code จะมีการเปลี่ยนบิต 2 บิต เช่นเดียวกับตำแหน่ง 3 เป็น 4 ก็จะมีการเปลี่ยนจำนวน 2 บิต เช่นเดียวกัน ซึ่งในการทำงานจริงมีโอกาสที่จะเกิดสัญญาณรบกวนและทำให้บิตเหล่านี้เปลี่ยนแปลง โดยจะทำให้เกิดปัญหาการปลอมบิตเกิดขึ้นจากค่าที่อ่านได้ เช่น 0001 ⟶ 0011 ⟶ 0010 (1 ⟶ 3 ⟶ 2) ซึ่งค่าที่ควรจะเป็นคือ 0001 ⟶ 0010 ⟶ 0011 (1 ⟶ 2 ⟶ 3) แต่สำหรับ Encoder Lika ที่ออกแบบมาพิเศษได้แก้ปัญหาเหล่านี้โดยอาศัยการเข้ารหัสแบบ Gray Code แทน ซึ่งดูได้จากรูปที่ 2.25 โดยจะเห็นได้ว่าใน Gray Code ตำแหน่ง (1 ⟶ 2 ⟶ 3) จะถูกเข้ารหัสด้วย 0001 ⟶ 0011 ⟶ 0010 แทนซึ่งถ้าเป็นใน Binary Code ค่านี้ถือว่าเป็นการเข้ารหัสผิด การเปลี่ยนแปลงของรหัสเกรย์แต่ละค่าจะต่างจากจำนวนก่อนหน้าอยู่ 1 บิตเสมอ ทำให้เกิดความผิดพลาดได้ยากเมื่อเทียบกับรหัสเลขฐานสองซึ่งอาจเกิดความคลาดเคลื่อนของรหัส (Erronous Code) หรือเกิดการคลุมเครือของรหัสระหว่างส่งรหัสได้ รหัส Gray Code จะสามารถจำกัดความผิดพลาดสูงสุดอยู่ที่ค่าของ LSB เสมอ

Decimal	0	1	2	3	4	5	6	7
Binary (4 digits)	0000	0001	0010	0011	0100	0101	0110	0111
Gray (4 digits)	0000	0001	0011	0010	0110	0111	0101	0100

รูปที่ 2.25 ภาพแสดงตารางเปรียบเทียบรหัส Gray Code กับ Binary Code

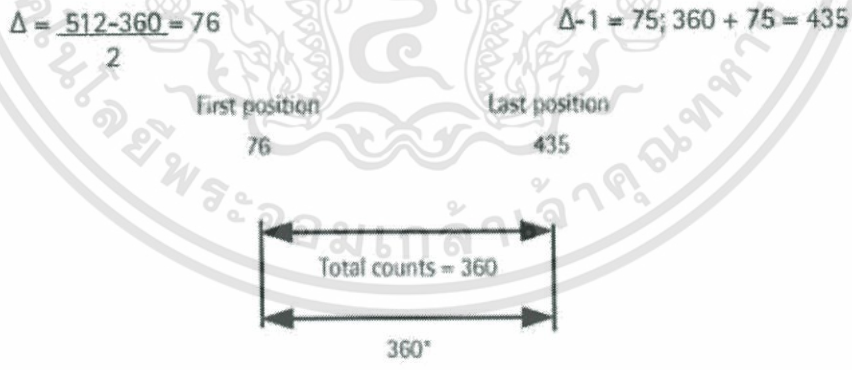
2.4.2.2 Shifted Gray Code or Gray Excess Code คือ การแปลงค่าจาก Binary Code ไปเป็นฐานสิบซึ่งจะสามารถแทนค่าสูงสุดได้เท่ากับ 2n โดยค่า n จะเป็นจำนวนบิตที่ใช้แทนค่า เช่น n = 9 บิต จะสามารถแทนค่าได้ 29 = 512 ซึ่งค่าเท่ากับ 512 โดยจะแทนค่ากลับเป็นตำแหน่งเดิมได้ ตัวอย่างเช่น Shifted Gray Code 360 counts/rev ค่า Integral Gray ของ Encoder = 512 ต้องการทราบว่าค่า First Position กับ Last Position จะต้องมีค่าเท่าไร โดยจากรูปที่ 2.26 จะแสดงให้เห็นว่าตำแหน่งเริ่มต้นของข้อมูลนั้นคือ 76 แทนตำแหน่ง 1 องศา และข้อมูลสุดท้ายคือ 435 แทนข้อมูลที่ตำแหน่ง 360 องศา ซึ่งสามารถคำนวณหาค่าดังกล่าวได้ดังนี้

Integral Gray – Shifted Gray2

Integral Gray = ค่าสูงสุดที่สามารถเกิดขึ้นได้ โดยหาได้จาก 2n

Shifted Gray = ค่าสูงสุดที่สามารถนับได้ต่อรอบ

= ค่าของข้อมูลชุดแรกที่แทนตำแหน่งเริ่มต้น -1 = ค่าของข้อมูลชุดสุดท้ายที่แทนตำแหน่งสุดท้าย



รูปที่ 2.26 การหาข้อมูลตำแหน่งของ Code

2.4.2.3 BCD Code หรือ Binary Coded Decimal เป็นการแทนเลขฐานสิบด้วยการใช้ BCD Code ซึ่งจะแทน 1 ชุด BCD = 4 บิต โดยแทนค่า 0 – 9 ซึ่งใน 4 บิตแต่ละบิตจะถูกแทนด้วยค่า 8 4 2 1 ในการหาค่านั้นให้เอาค่ามาแทนในบิตที่มีค่าเป็น 1 แล้วเอาค่ามารวมกัน เช่น 0001 0110 = 1 6 แต่ถ้าเป็น Binary Code จะได้ค่า 22

2.4.2.4 Output Circuits ในส่วนของวงจร Output จะเป็นเรื่องของระดับสัญญาณและวิธีการเชื่อมต่อกันทางสัญญาณไฟฟ้าระหว่างตัวเซนเซอร์กับตัวอ่านค่าไม่ว่าจะเป็นตัวนับจำนวนหรือระบบคอนโทรลเลอร์ PLC, DCS ซึ่งใน Encoder แต่ละรูปแบบจะมีสัญญาณ Output ที่เหมือนกันหรือต่างกันก็ได้ โดยวงจร Output ของ Encoder ที่มีใช้กันอยู่ในปัจจุบันได้แก่ NPN Open Collector (Code N) เป็น Output แบบ NPN หรือ Sinking Output จะใช้งานกับอุปกรณ์ที่ทำหน้าที่จ่ายกระแสออกมา ซึ่งปกติวงจร NPN Open Collector จะถูกใช้บ่อยในกรณีที่อุปกรณ์ที่นำมาต่อนั้นมีแหล่งจ่ายแรงดันไม่เท่ากับแหล่งจ่ายแรงดันของ Encoder โดยภายในวงจรของ Encoder จะมีทรานซิสเตอร์ชนิด NPN อยู่โดยส่วนอุปกรณ์ที่นำมาต่อด้วยนั้นจำเป็นต้องมีตัวต้านทานจำกัดกระแสไว้ หรือ R Pull-Up เพื่อป้องกันการลัดวงจร

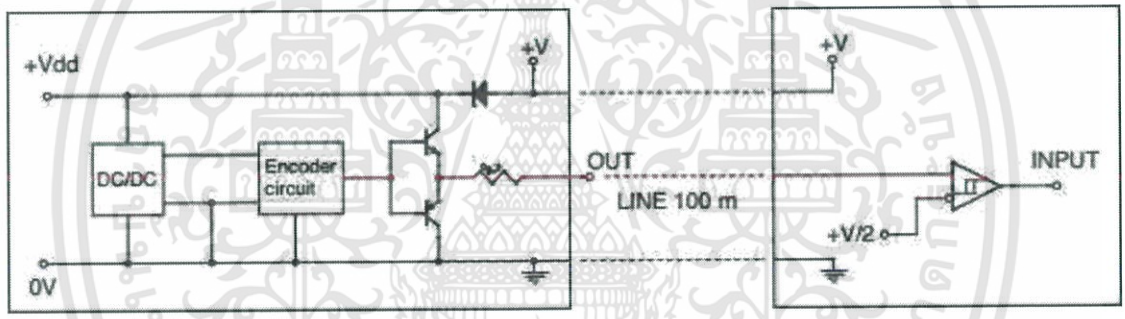
2.4.2.5 PNP open collector (Code P) เป็น Output แบบ PNP หรือ Sourcing Output จะใช้งานกับอุปกรณ์ที่ทำหน้าที่เป็นโหลดโดยตัว Encoder จะทำหน้าที่จ่ายกระแสให้กับอุปกรณ์ ซึ่ง Encoder แบบนี้จะเป็นชนิดที่ไม่ค่อยมีใช้งาน โดยรูปที่ 2.27 เป็นคุณสมบัติของ Output

Power supply (Vin)	Max. output current	Peak output current	Typical rising/falling edge time	Vout low @40mA	Vout high @40mA
+10 +30 Vdc	40 mA	80 mA	550 ns / 470 ns	acc. to following electronics	(Vin – 1,25) Vdc

รูปที่ 2.27 ภาพแสดงคุณสมบัติของวงจร Output แบบ PNP Open Collector

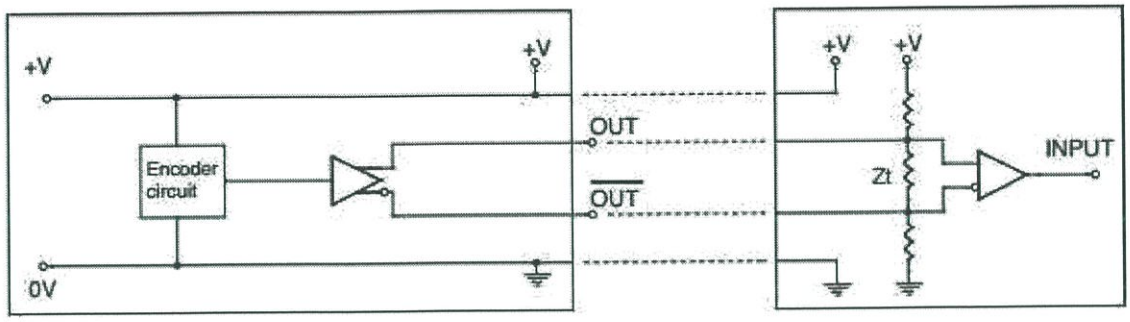
2.4.2.6 Push-Pull/HTL (Code Y) เป็นวงจร Output ของ Encoder ซึ่งจะสร้างสัญญาณรูปคลื่นสี่เหลี่ยมซึ่งมีมุมเฟสต่างกันอยู่ที่ 90 องศา ซึ่งวงจร Push-Pull นั้นจะสามารถรองรับสัญญาณการทำงานได้ทั้งแบบ Sinking และ Sourcing โดยทนกระแสได้สูงถึง 80 mA ต่อ Output ซึ่งสัญญาณ Output แบบนี้จะเหมาะกับอุปกรณ์ที่เป็น Motion Control และอุปกรณ์ที่กินกระแสไม่เยอะมาก

2.4.2.7 Power Push-Pull/HTL (Code T) เป็นวงจร Output ของ Encoder ดังรูปที่ 2.28 ซึ่งเหมือนกับวงจร Output แบบ Push-Pull/HTL (Code Y) แต่จะมีคุณสมบัติที่สามารถขับโหลดที่มีระยะทางไกลได้ โดย Output แบบนี้ถูกออกแบบมาสำหรับแก้ปัญหาของวงจร Output แบบ Push-Pull/HTL (Code Y) ที่ไม่สามารถต่อสายในระยะไกลได้และสำหรับวงจร Output แบบ Power Push-Pull/HTL นั้นสามารถต่อสายเซนเซอร์ได้ไกลถึง 100 เมตร



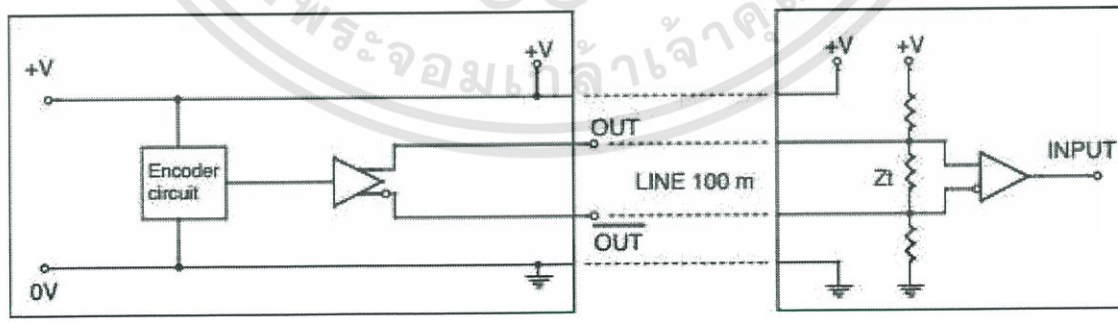
รูปที่ 2.28 การต่อวงจร Output แบบ Power Push-Pull/HTL (Code T)

2.4.2.8 Line Driver (RS-422)/TTL (Code L) เป็นวงจร Output ของ Encoder ที่ใช้ IC 26LS31 ดังรูปที่ 2.29 ซึ่งเป็นวงจรแบบ Sourcing เหมาะสำหรับการต่อสาย Encoder ที่มีระยะทางไกล ซึ่งอาจมีสัญญาณรบกวนเกิดขึ้นได้ง่าย โดยวงจรแบบนี้ออกแบบมาเพื่อป้องกันระบบสื่อสารให้สามารถทำงานได้อย่างถูกต้อง ด้วยวงจรนี้จะให้สัญญาณแบบ Inverted AB0/AB0 ที่เป็นพัลส์ ซึ่งในการใช้งานจริงของสัญญาณนั้น จะมีเรื่องรบกวนที่เกิดขึ้นได้ทั้งฝั่งที่เป็นสัญญาณปกติ (Normal Signal) และสัญญาณกลับ (Inverted Signal) ซึ่งสัญญาณสัญญาณรบกวนเหล่านี้ สามารถจัดการได้ง่ายโดยวงจรภาครับแบบ Differential ซึ่งเป็นภาครับโดยทั่วไปของตัวคอนโทรลเลอร์อยู่แล้ว ซึ่งระดับแรงดันที่ใช้ในวงจรนี้จะเป็นแบบ 5 โวลต์ DC โดยอ้างอิงตนามมาตรฐาน EIA RS-422



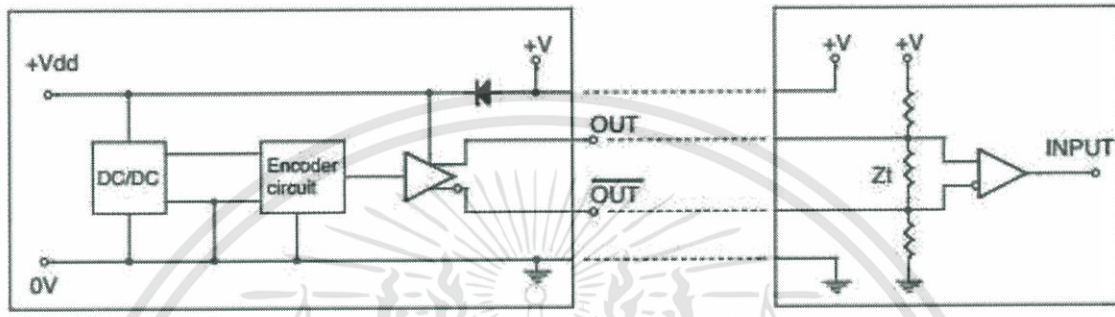
รูปที่ 2.29 ภาพการต่อวงจร Output แบบ Line Driver (RS-422)/TTL (Code L)

2.4.2.9 Power Line Driver (RS-422)/TTL (Code K) เป็นวงจร Output ของ Encoder แบบ Sourcing ดังรูปที่ 2.30 เหมาะสำหรับการต่อสาย Encoder ที่มีระยะทางไกลซึ่งอาจมีสัญญาณรบกวนเกิดขึ้นได้ง่าย โดยวงจรนี้จะมีการปรับปรุงภาคกำลังส่งให้สามารถขับกระแสได้สูงถึง 300mA เพื่อปรับปรุงระยะทางในการส่งให้ดีกว่าวงจรแบบ Line Driver (RS-422)/TTL (Code L) และยังใช้เทคนิคการใช้ค่าความต้านทาน 75 โอห์ม เพื่ออาศัยหลักการ Power Maximum Transfer จึงทำให้ได้กำลังส่งสูงสุด โดยสัญญาณ Output จะยังคงเป็นแบบ Inverted AB0/AB0 ที่เป็นพัลส์ ซึ่งในการใช้งานจริงนั้นจะมีสัญญาณรบกวนเกิดขึ้นทั้งฝั่งที่เป็นสัญญาณปกติ (Normal Signal) และสัญญาณกลับ (Inverted Signal) ซึ่งสัญญาณรบกวนเหล่านี้สามารถจัดการได้ง่ายโดยวงจรภาครับแบบ Differential ซึ่งเป็นภาครับโดยทั่วไปของตัวคอนโทรลเลอร์อยู่แล้ว โดยระดับแรงดันที่ใช้ในวงจรนี้จะเป็นแบบ 5 โวลต์ DC โดยอ้างอิงตนามมาตรฐาน EIA RS-422



รูปที่ 2.30 การต่อวงจร Output แบบ Power Line Driver (RS-422)/TTL (Code K)

2.4.2.10 Universal Circuit Push-Pull + Line Driver/HTL + TTL (Code H) รูปที่ 2.31 วงจรภาค Output แบบนี้ได้ถูกพัฒนาและออกแบบ โดยมีเอกสิทธิ์เฉพาะ Lika ซึ่งรวบรวมข้อดีทั้งของวงจรแบบ Push-Pull และ Line Driver ไว้ด้วยกัน ซึ่งแทบจะครอบคลุมมาตรฐานของผู้ผลิต Encoder ทั่วโลก



รูปที่ 2.31 การต่อวงจร Output แบบ Universal Circuit Push-Pull+Line Driver/HTL+TTL(Code H)

2.4.2.11 1 Vpp sin/cos (Code V) คือ วงจรภาค Output ของ Encoder ที่ให้สัญญาณ Output ที่เป็นแบบแรงดันไฟฟ้าโดยมีมุมเฟสต่างกันอยู่ที่ 90 องศา ระหว่างเฟส A และ B โดยเฟส A เป็นฟังก์ชัน Sine และ B เป็นฟังก์ชัน Consine โดยค่าระดับแรงดันไฟฟ้าจะมีขนาด Amplitude อยู่ที่ 0.5 โวลต์ pp โดยจะอยู่บนแรงดัน Offset 2.5 โวลต์ ซึ่งแรงดัน 1 Vpp ได้มาจากค่าแรงดันระหว่างเฟส A และเฟส /A โดยความถี่ของสัญญาณนั้น จะขึ้นอยู่กับความเร็วของการหมุนของEncoder ถ้าหมุนเร็วความถี่ก็จะสูงขึ้น

2.4.2.12 Analogue Output เป็นสัญญาณ Output ของ Encoder ที่นิยมเอามาใช้ในวงจรป้อนกลับในเรื่องของความเร็ว หรือตำแหน่งของ Encoder เช่น งานควบคุมความเร็ว หรือตำแหน่งของมอเตอร์ โดยโครงสร้างภายในของ Encoder จะทำหน้าที่แปลงค่าที่อ่านได้จาก Code Disk ให้เป็นสัญญาณมาตรฐาน เช่น 0 to 5 โวลต์, 0 to 10 โวลต์, -5 to +5 โวลต์, -10 to +10 โวลต์, 4 to 20 มิลลิแอมป์, 0 to 20 มิลลิแอมป์ และ 0 to 24 มิลลิแอมป์

2.4.2.13 Fieldbus Interface (Serial Interfaces) เป็นมาตรฐานการสื่อสารระหว่างอุปกรณ์ภาคสนาม เช่น เซนเซอร์ต่างๆ กับส่วนที่เป็นตัวประมวลผล เช่น หน้าจอแสดงผล

ต่างๆ พีแอลซี ดีซีเอส โดยระบบสื่อสารใน Fieldbus จะเป็นการส่งข้อมูลแบบอนุกรม และเป็นสัญญาณแบบดิจิตอล โดยมีมาตรฐานและการทำงานแต่ละแบบที่แตกต่างกัน โดยสามารถเปรียบเทียบตัวอย่าง Field Bus ต่างๆ ได้ดังตารางที่ 2.2

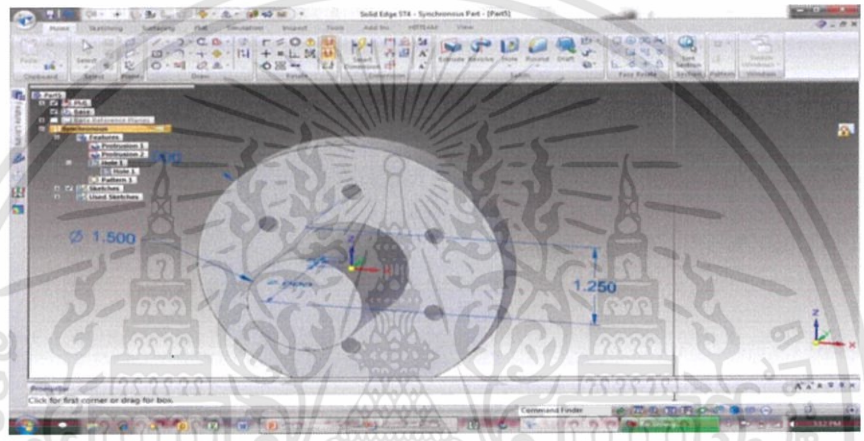
ตารางที่ 2.2 เปรียบเทียบตัวอย่าง Field Bus

Bus Technology	Standard	Bus Power	Communication Speed	Std. Max. Distance	Max. Devices
AS-Interface	AS-International	Yes	-	100m	62
CANopen	CiA 301, CiA 401	No	1Mbps at 20m	5km at 10kbps	127
CC-Link	IEC 61784, CLPA	No	10Mbps	1.2km	64
DeviceNet	ODVA, CIP	Yes	125, 250, 500kbps	500m	64
EtherCAT	IEC 61158, IEEE 802.3	No	100Mbps, 1Gbps	100m	65,536
Ethernet/IP	ODVA, CIP	No	100Mbps, 1Gbps	100m	Unlimited
FF H1	IEC 61158, ISA SP50	Yes	31.25Kbps	1.9 – 9.5km	240
FF HSE	IEC 8802, IEEE 802.3	No	100Mbps, 1Gbps	100m	Unlimited
MODBUS	TIA-485	No	9.6Kbps – 12Mbps	1.5km	246
PROFIBUS PA	IEC 61158	Yes	31.25Kbps	1.9 – 9.5km	126
PROFIBUS DP	IEEE1451.2, TIA-485	No	9.6Kbps – 12Mbps	1.5km	126
PROFINET	IEC 8802, IEEE 802.3	No	100Mbps, 1Gbps	100m	Unlimited
HART	Bell 202, 4..20mA	Yes	1.2 Kbps – 9.6Kbps	3km	64

2.4.2.14 BiSS Interface หรือ Bidirectional Serial and Synchronous เป็นการสื่อสารแบบดิจิตอลแบบหนึ่ง ถูกพัฒนาขึ้นมาในปี 2002 เพื่อใช้กับตัว Sensor และ Actuator ซึ่งมีการสื่อสารแบบ 2 ทิศทาง เชื่อมต่อสัญญาณแบบอนุกรมและใช้สัญญาณนาฬิกาแบบ Synchronous ในการให้จังหวะในการส่งข้อมูล ในปัจจุบันยังมีการใช้งานการสื่อสารประเภทนี้อยู่ เนื่องจากความสามารถในการส่งข้อมูลที่มีความเร็วสูง และมีความทนต่อการรบกวนของสัญญาณจากสนามแม่เหล็กได้เป็นอย่างดี

2.5 Solid Edge

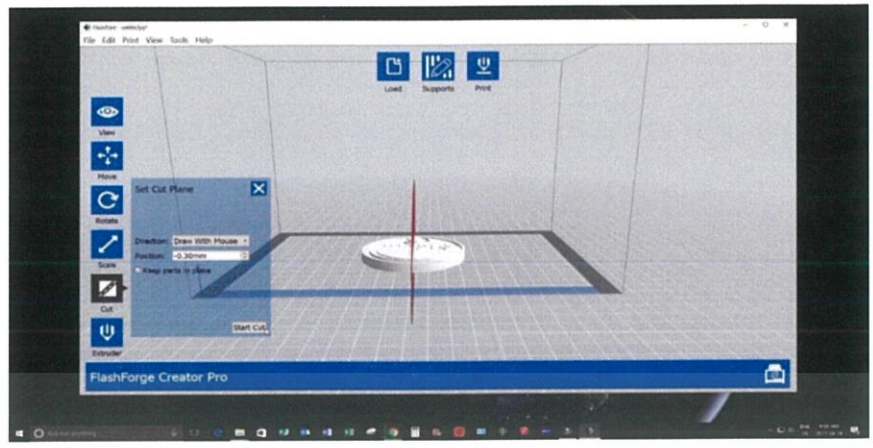
Software Solid Edge ST10 โปรแกรมออกแบบภาพ 3D จากค่าย SIEMENS PLM ที่ได้นำเอา Synchronous Technology มาช่วยในเรื่องของการออกแบบงานต่างๆ ให้ง่ายและรวดเร็วมากยิ่งขึ้น แถมยังสามารถแก้ไขไฟล์ 3D จากซอฟแวร์อื่นๆ ได้อย่างรวดเร็ว อีกทั้งฟังก์ชันการ Reverse Engineering ที่ทำให้การปรับแต่งหรือแก้ไขงานที่ได้จากเครื่องสแกนเนอร์ 3D ต่างๆ ดังรูปที่ 2.32



รูปที่ 2.32 หน้าจอโปรแกรม Solid Edge

2.6 Flashprint

เป็นโปรแกรมประเภท Slicer ซึ่งหมายความว่าตัวโปรแกรมจะทำการตัดโมเดล 3D ออกเป็นแผ่นบางๆ แล้วแปลงเป็นไฟล์ประเภท Gcode ซึ่งเป็นภาษาที่เครื่องพิมพ์ 3D เข้าใจเพื่อส่งไปให้เครื่องพิมพ์จัดการพิมพ์เป็นชิ้นงานออกมา ดังรูป 2.33 และตารางที่ 2.3



รูปที่ 2.33 หน้าจอโปรแกรม Flashprint

ตารางที่ 2.3 คำสั่งต่างๆ ของโปรแกรม Flashprint

	Load	โหลดไฟล์โมเดล
	Support	เข้าสู่โหมดสร้าง Support
	View	เปลี่ยนมุมมองของหน้าจอ
	Move	ย้ายตำแหน่งโมเดล
	Rotate	หมุนโมเดลได้ทั้งสามแกน
	Scale	ย่อ-ขยายโมเดล

ตารางที่ 2.3 คำสั่งต่างๆ ของโปรแกรม Flashprint (ต่อ)

	Cut	ตัดโมเดลในทิศทางต่างๆ ตามต้องการ
	Print	สั่งพิมพ์โมเดลที่อยู่บนหน้าจอ

2.7 งานวิจัยที่เกี่ยวข้อง

Blue Robot ติดตั้งใช้งานอยู่ที่ บริษัท เด็นโซ่ (ประเทศไทย) จำกัด (โรงงานบางปะกง)

บทที่ 3

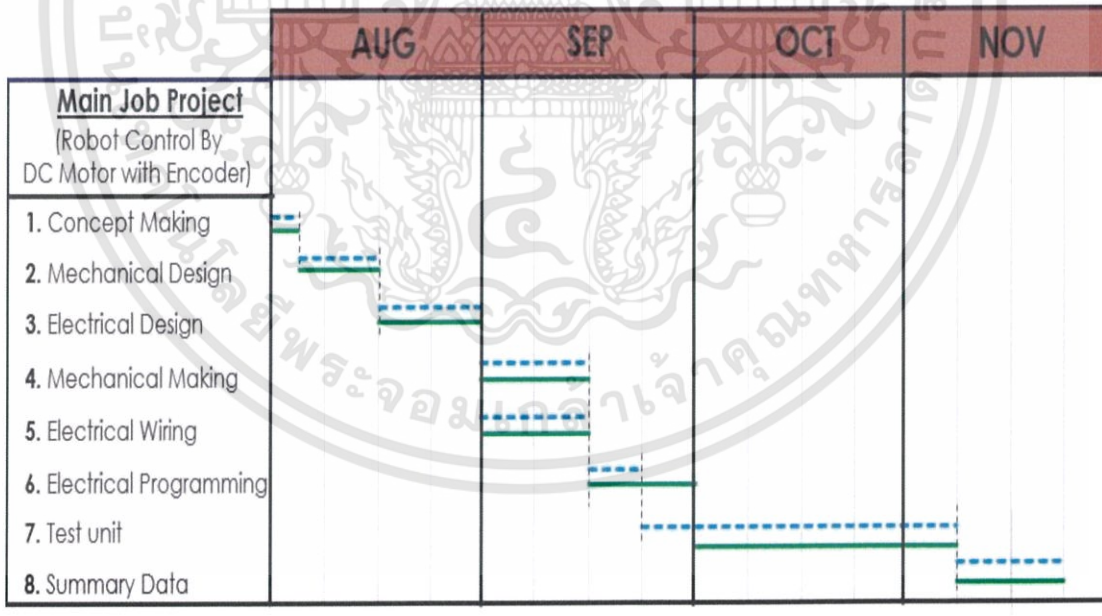
วิธีการดำเนินโครงงาน

การดำเนินโครงงานแขนกลด้วย DC Motor เริ่มดำเนินทำโครงงานตั้งแต่ การออกแบบโครงสร้างทางกล การออกแบบวงจรไฟฟ้ารวมถึงการประกอบและเขียนโปรแกรม หลังจากเขียนโปรแกรมสั่งงานแล้วจะเริ่มทำการทดสอบแขนกลเพื่อทดสอบความเที่ยงตรง

3.1 การวางแผนการดำเนินงาน

ในการจัดทำโครงงานจะต้องมีการวางแผนงานเป็นขั้นตอน และจัดลำดับช่วงเวลาของแต่ละส่วนงาน เพื่อให้สามารถดำเนินงานได้เป็นระบบ โดยแผนงานที่ได้วางแผนไว้เป็นช่วงเวลาที่ดำเนินโครงการสหกิจศึกษา ณ บริษัท เด็นโซ่ (ประเทศไทย) จำกัด (โรงงานบางปะกง) คือช่วงเวลาระหว่างวันที่ 6 สิงหาคม 2561 ถึงวันที่ 30 พฤศจิกายน 2561 แสดงแผนการดำเนินงานดังตารางที่ 3.1

ตารางที่ 3.1 DC Robot Arm Tmeline



3.2 การศึกษาโครงสร้างและการทำงานของแขนกลที่มีอยู่เดิม

ก่อนที่จะออกแบบแขนกลที่สามารถใช้งานได้จริง และตรงตามข้อกำหนดที่ตั้งไว้จะต้องทราบก่อนว่าแขนกลที่ใช้อยู่ในปัจจุบันมีหลักการทำงานอย่างไร เพื่อนำข้อมูลการทำงานไปใช้ในการออกแบบตัวโครงสร้างทางกลของแขนกล การออกแบบวงจรไฟฟ้า รวมถึงการออกแบบโปรแกรมควบคุมการทำงาน

นอกจากศึกษาโครงสร้างของแขนกลเดิมแล้ว ยังต้องศึกษาสภาพแวดล้อมรอบๆ แขนกลเพื่อเพื่อใช้เป็นข้อมูลในการออกแบบโครงสร้างอื่นๆ ต่อไป

3.3 การออกแบบโครงสร้างและชิ้นส่วนทางกล

ในการออกแบบโครงสร้างและชิ้นส่วนทางกลของแขนกล ต้องมีการอ้างอิงจากขนาดแขนกลที่ใช้อยู่เดิม แขนกลที่นำมาใช้คือ Blue Robot โดย Blue Robot เป็นแขนกลที่พนักงานในแผนก R&D สั่งทำขึ้นมาเอง

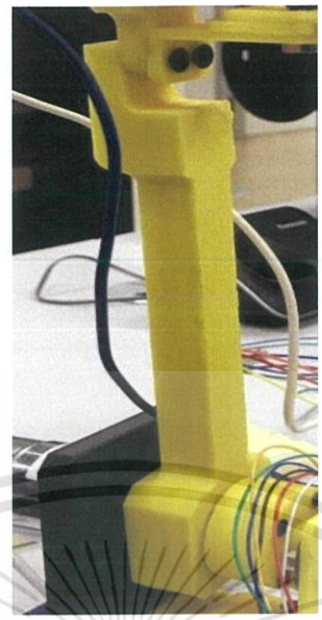
หลังจากทำการเลือกแขนกลที่ใช้แล้วก็จะนำขนาดชิ้นส่วนของแขนกลที่ใช้มาเป็นข้อมูลใน การออกแบบโครงสร้างและชิ้นส่วนทางกลต่างๆ

โดยมีชิ้นส่วนต่างๆ ดังนี้

3.3.1 โครงสร้างตัวแขน ดังรูปที่ 3.1

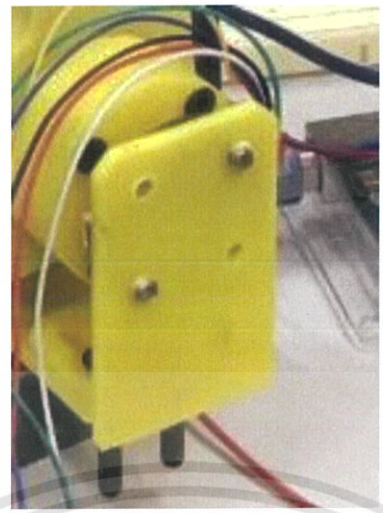
3.3.2 โครงสร้างฐานวางแขนกล ดังรูปที่ 3.2

3.3.3 โครงสร้างตัวยึด Encoder ดังรูปที่ 3.3



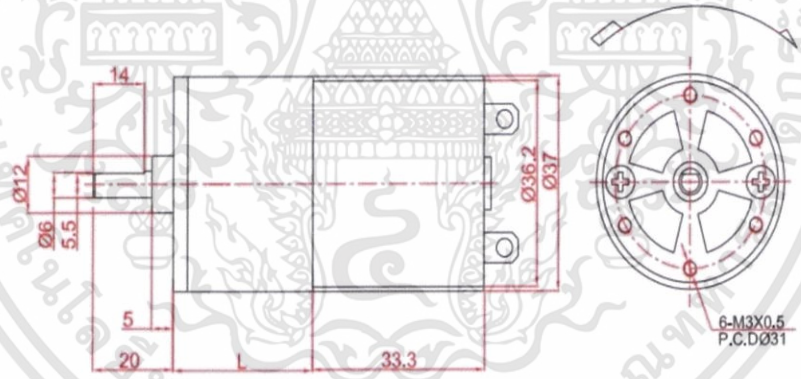
รูปที่ 3.1 โครงสร้างตัวแขน

รูปที่ 3.2 โครงสร้างฐานวางแขนกล

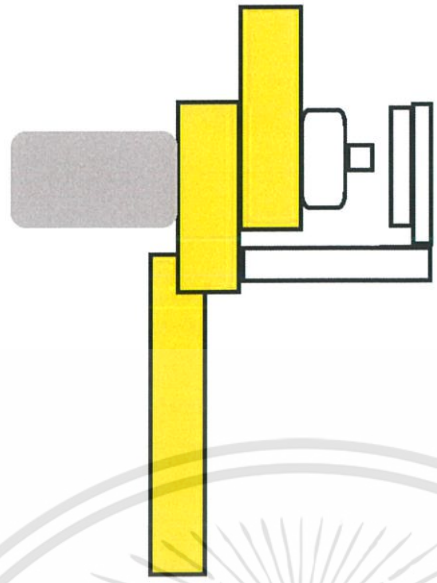


รูปที่ 3.3 โครงสร้างตัวยึด Encoder

โดยตัวฐานและแขนจะออกแบบมาให้มีขนาดที่เหมาะสมกับ SMG:Gear Motor 24VDC ดังรูปที่ 3.4 และรูปที่ 3.5



รูปที่ 3.4 SMG:Gear Motor 24VDC



รูปที่ 3.5 โครงสร้างแขนกล

3.4 การออกแบบโปรแกรม

3.4.1 Motor Control

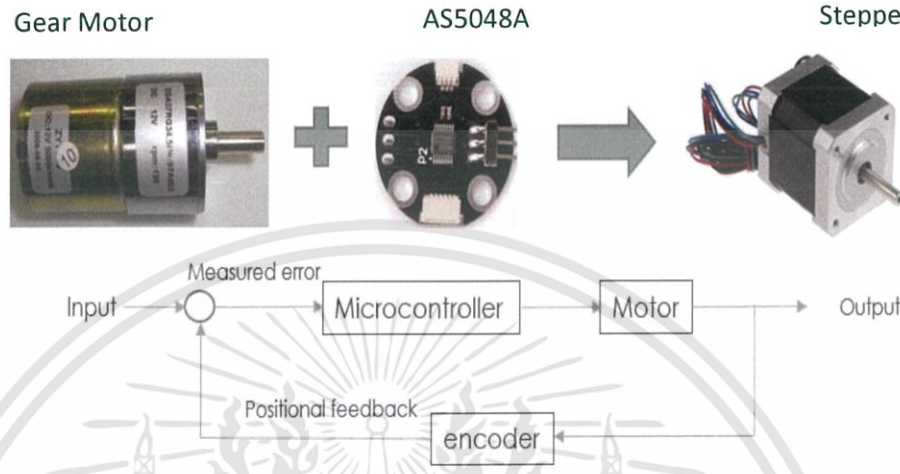
โปรแกรมในส่วนนี้เป็นตัวสั่งการให้ Motor ขยับไปข้างหน้าหรือข้างหลังตามที่สั่งการ โดยตัวโปรแกรมจะทำงานไปพร้อมๆ กับโปรแกรมควบคุม Encoder

3.4.2 Encoder Control

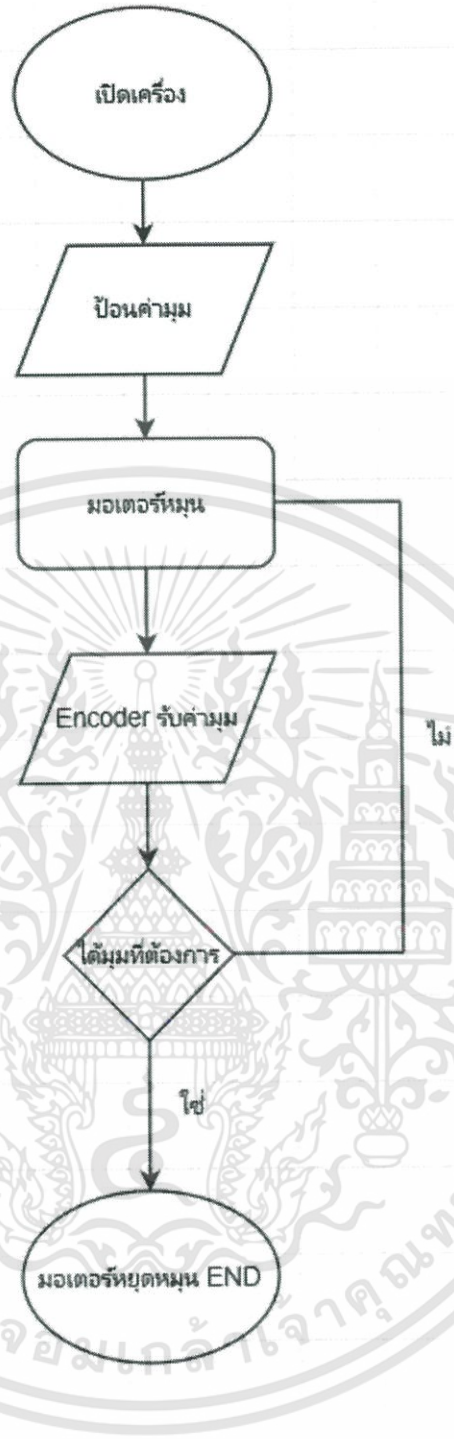
โปรแกรมในส่วนนี้เป็นตัวเปิดการทำงานของ Encoder และให้ Encoder รับค่าตำแหน่งของแกน Motor หลังจากนั้นก็ส่งค่ากลับไปยังตัวบอร์ด Arduino เพื่อให้ Arduino สั่งการกลับไปที่โปรแกรม Motor Control

จากรูปที่ 3.5 Strpping Motor (สเต็ปปิ้งมอเตอร์) หรือ Stepper Motor (สเต็ปเปอร์มอเตอร์) คือ มอเตอร์ไฟฟ้าที่ขับเคลื่อนด้วยพัลส์มีลักษณะการขับเคลื่อนโดยการหมุนรอบแกน 360 องศา แต่มีลักษณะไม่ต่อเนื่องจะเคลื่อนไปเป็นสเต็ปโดยแต่ละสเต็ปจะขับเคลื่อนได้ 1, 1.5, 1.8 หรือ 2 องศา ซึ่งจะขึ้นอยู่กับแต่ละโครงสร้างของมอเตอร์

ซึ่งโดยปกติแล้ว Gear Motor จะไม่สามารถทำงานได้แบบ Stepper Motor ทางผู้ใช้งานจึงได้ทำการนำอุปกรณ์ที่สามารถส่งค่ากลับได้อย่าง Encoder มาใช้งานร่วมกับ Gear Motor ดังรูปที่ 3.6 และสามารถนำมาเขียน Flowchart ได้ดังรูปที่ 3.7



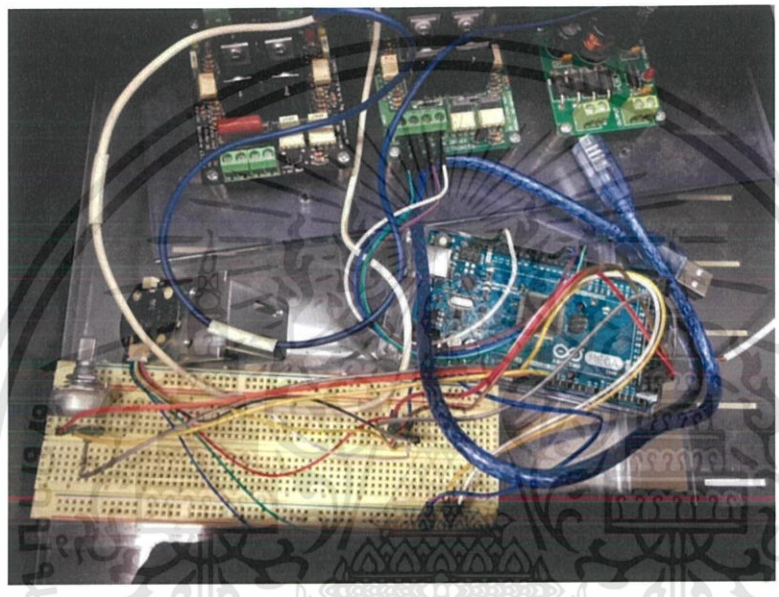
รูปที่ 3.6 รูปอธิบายหลักการทำงาน

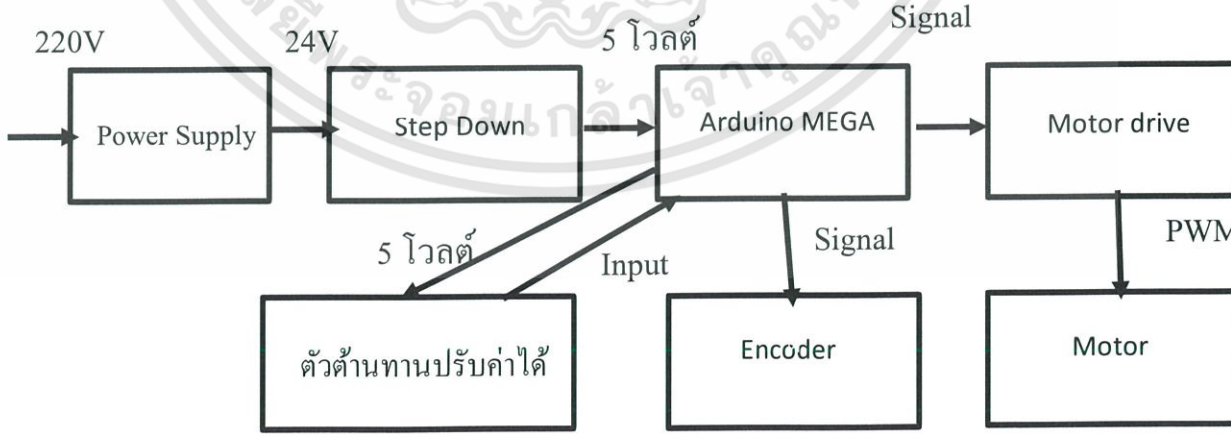


รูปที่ 3.7 Flowchart แสดงการทำงาน

3.5 ส่วนวงจรไฟฟ้า

ในส่วนของระบบควบคุมจะใช้บอร์ด Arduino MEGA โดยการเขียน Code ผ่านโปรแกรม ArduinoIDE แล้วทำการอัพโหลดลงไปในบอร์ด Arduino MEGA ซึ่งจะมีลักษณะการต่อวงจรดังรูปที่ 3.8 โดยจากรูปจะเป็นการต่อวงจรควบคุมมอเตอร์หนึ่งตัวซึ่งกรณีที่ต้องการควบคุมมอเตอร์สามตัวก็ต้องใช้วงจรนี้สามชุด





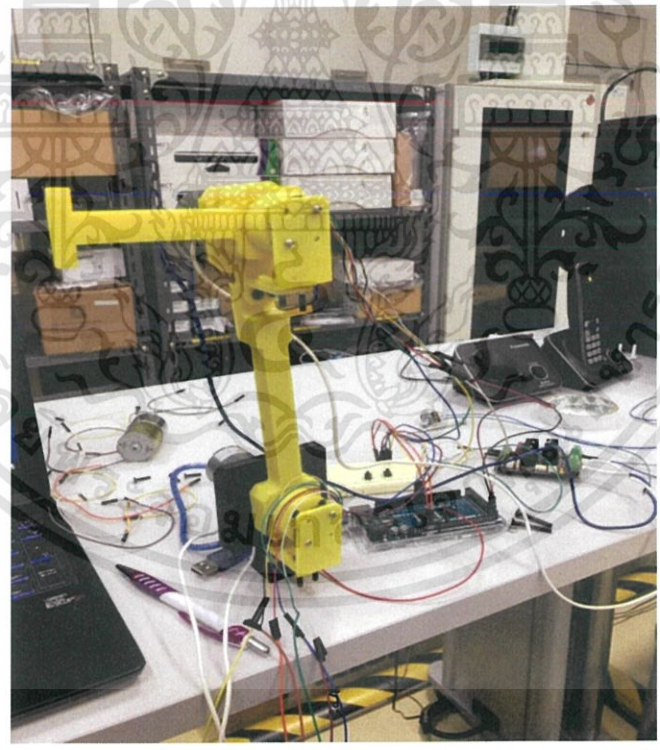
รูปที่ 3.8 วงจรแขนกล

บทที่ 4

ผลการดำเนินโครงงาน

4.1 โครงสร้างเครื่อง

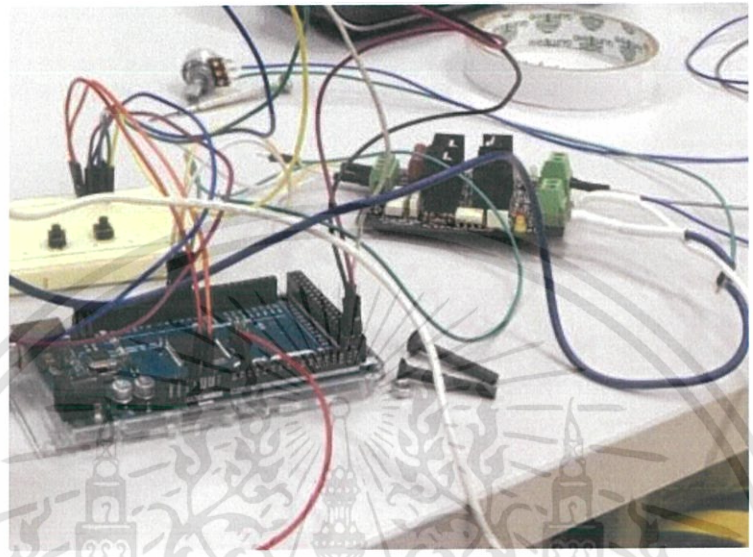
ส่วนประกอบทางกลของแขนกลถูกดำเนินการประกอบจนเสร็จสิ้น 100 เปอร์เซ็นต์ โดยโครงสร้างแขนกล ถูกแบ่งออกเป็น 3 ส่วน ได้แก่ ส่วนแขน ส่วนฐาน และส่วนยึด Encoder โดยทำการเริ่มประกอบในส่วนของฐานก่อนเป็นอันดับแรก เนื่องจากเป็นส่วนที่อยู่ล่างสุดของแขนกล ส่วนต่อมาที่ทำการกระกอบต่อมาคือ ส่วนแขน โดยทำการยึดแขนเข้ากับ Motor ที่ติดอยู่กับส่วนฐาน ในส่วนสุดท้ายเป็นส่วนของ Encoder ซึ่งเป็นส่วนที่ประกอบยากมากที่สุดเนื่องจากต้องประกอบให้ Encoder กับแม่เหล็กแผ่นกลมที่ติดอยู่กับปลายมอเตอร์ สัมผัสกันพอดีเพื่อให้ได้ค่า Feedback ที่แม่นยำ ดังรูปที่ 4.1



รูปที่ 4.1 แขนกลที่ประกอบเสร็จแล้ว

4.2 ส่วนวงจรไฟฟ้า

ผลการดำเนินงานในด้านส่วนประกอบทางไฟฟ้าสำเร็จไป 100 เปอร์เซ็นต์ ระบบไฟฟ้าของแขนกลทำงานได้ตามปกติ ไม่มีอุปกรณ์ชิ้นใดเสียหายหรือลัดวงจร ดังรูปที่ 4.2



รูปที่ 4.2 ตัวอย่างการเชื่อม Arduino กับ Motor Drive

4.3 ผลการทดสอบโปรแกรม

ทำการทดสอบโปรแกรม โดยการขยับลูกบิดเพื่อทดสอบการทำงาน และลำดับการทำงานของแขนกลว่าเป็นไปตามลำดับการทำงานที่กำหนดไว้หรือไม่

จากการทดสอบโปรแกรมควบคุมลำดับการทำงานผู้ดำเนินโครงงานพบว่า แขนกลสามารถขยับได้เที่ยงตรงในระดับหนึ่ง ยังมีค่าความคลาดเคลื่อนประมาณ 5 องศา เนื่องจากตัวโครงสร้างของแขนกลจำลองทำมาจากวัสดุพลาสติกซึ่งโครงสร้างไม่แข็งแรงเท่าที่ควร ทำให้เมื่อแขนกลขยับตัวแขนกลจะสั่น Encoder จึงอ่านค่าได้คลาดเคลื่อนจากที่กำหนดไว้เล็กน้อย ซึ่งถือว่าบรรลุเป้าหมายในการทำงานวิจัยโดยสามารถลดต้นทุนของแขนกลและสามารถลดความซับซ้อนของระบบการทำงาน ดังรูปที่ 4.3 และรูปที่ 4.4

```
void enc()
{
  digitalWrite(SS, LOW);
  result1 = SPI.transfer(0b00000000);
  result1 &= 0b00111111;
  result1 = result1 << 8;
  result2 = SPI.transfer(0b00000000);
  result = result1 | result2;
  digitalWrite(SS, HIGH);
  e = map(result, 0, 16383, 0, 360); //นำค่าจาก encoder มาแปลงเป็นองศา
```

รูปที่ 4.3 ชุดคำสั่งของ Encoder

```
void GoToAngle(int target, int encode)
{

  int current = encode;
  Serial.print("current = ");
  Serial.println(current);
  Serial.print("target = " );
  Serial.println(target);
  if(target>current&&target-current!=PERM_ERROR&&target-current>PERM_ERROR)
  {
    turnleft();
   Serial.println("turnleft");
}
  else if(target<current&&current-target!=PERM_ERROR&&current-target>PERM_ERROR)
  {


   turnright();
    Serial.println("turnright");
}
  else if(target==current||target-current==PERM_ERROR||current-target==PERM_ERROR||0<current-target<PERM_ERROR||0<target-current<PERM_ERROR)
  {
    Stop();
    delay(50);
  }
```

รูปที่ 4.4 ตัวอย่างชุดคำสั่งของ Motor

บทที่ 5

สรุปผลและข้อเสนอแนะ

โครงงานฉบับนี้ได้นำเสนอการออกแบบแขนกลด้วยเกียร์มอเตอร์ของทาง บริษัท เด็นโซ่ (ประเทศไทย) จำกัด (โรงงานบางปะกง) โดยมีวัตถุประสงค์ในการแก้ไขปัญหาเรื่องต้นทุนในการผลิตแขนกล และต้องการลดความซับซ้อนของระบบการทำงาน

5.1 สรุปผลการดำเนินงาน

จากการดำเนินงานโครงงานแขนกลด้วยมอเตอร์ DC นั้นแขนกลสามารถขยับได้เที่ยงตรงในระดับหนึ่ง ยังมีค่าความคลาดเคลื่อนประมาณ 5 องศา เนื่องจากตัวโครงสร้างของแขนกลจำลองทำมาจากวัสดุพลาสติกซึ่งโครงสร้างไม่แข็งแรงเท่าที่ควร ทำให้เมื่อแขนกลขยับตัวแขนกลจะสั่น Encoder จึงอ่านค่าได้คลาดเคลื่อนจากที่เป็น ซึ่งถือว่าบรรลุเป้าหมายในการทำงานวิจัยโดยสามารถลดต้นทุนของแขนกลและสามารถลดความซับซ้อนของระบบการทำงาน

5.2 ข้อเสนอแนะ

1. การวางแผนการดำเนินงาน มีความสำคัญต่อการทำโครงการเป็นอย่างมาก ต่อการทำงานต้องมีการวางแผนการดำเนินงานเผื่อระยะเวลาไว้ เนื่องจากอาจเกิดปัญหาในการดำเนินงานที่ทำให้ใช้ระยะเวลาในการดำเนินงานมากขึ้น

2. ควรสื่อสารงานให้เข้าใจภายในกับทีมปฏิบัติงาน หากสงสัยหรือไม่มั่นใจในงานที่ได้รับมอบหมายต้องสอบถามให้เข้าใจและชัดเจน เพื่อไม่ให้เกิดการทำงานที่ผิดพลาด ซึ่งจะส่งผลให้เกิดปัญหาตามมาในภายหลัง

3. ความเต็มที่กับงานเป็นสิ่งสำคัญ เพราะแสดงให้เห็นถึงความตั้งใจในการทำงาน

เอกสารอ้างอิง

[1] "**AS5048 Arduino**" (Online). Available :
https://github.com/ZoetropeLabs/AS5048A-Arduino

[2] "**AS5048A Reading Data**" (Online). Available :
https://forum.arduino.cc/index.php?topic=155238.0

[3] "**มอเตอร์ไฟฟ้า**" (Online). Available :
http://www.sangtawan.org/product_detail.asp?product_id=38&lng=th

[4] "**Step Down Converter**" (Online). Available :
https://www.digikey.com/en/articles/techzone/2014/oct/synchronous-step-down-dc-dc-converters-handle-wide-input-voltage-ranges?utm_adgroup=&mkwid=s&pcrid=313674480163&pkw=&pmt=b&pdv=c& productid=&slid=&pgrid=66131407401&ptaid=dsa-481824849274&gclid=CjwKCAiAyrXiBRAjEiwATI95mcRYKa3JxUwnlAO__D-c6T6bwS6G9u60BGihxh_2aNgTTxKyE9KY8xoC4T8QAvD_BwE

[5] "**Solid Edge Tutorial**" (Online). Available :
https://community.plm.automation.siemens.com/t5/Solid-Edge-Forum/Solid-Edge-best-Tutorials/td-p/433795

ภาคผนวก

ภาคผนวก ก.

โปรแกรมควบคุม Motor

```
#include "SPI.h"

#include <avr/io.h>

#include <util/delay.h>

char angle;

char a[20];

char b[20];

int a_size = 0;

int b_size = 0;

int data1 = 0;

int data2 = 0;

int enA = 5;

int in1 = 3;

int in2 = 4;

int e;

int turnSpeed = 30;

int currentPosition;

int currentPosition2;
```

```
unsigned int result = 0;

unsigned int result1 = 0;

unsigned int result2 = 0;

int turnDirection;

int PERM_ERROR = 10;

int in = 1;

int index = 1;

int maxAngle = 360;

int minAngle = 0;

int maxForwardBackwardAngle = 160;

int minForwardBackwardAngle = 20;

int maxUpDownAngle = 230;

int minUpDownAngle = 20;

int
readUpDown,readForwardBackward,readRotate,readGrab,readTeach,readStart,read
Stop,readUpDown_Grib,readRotate_Grib;

int
teachUpDown[100],teachForwardBackward[100],teachRotate[100],teachGrab[100],t
eachUpDown_Grib[100],teachRotate_Grib[100];

boolean started =false;
```

```
int StartPositionForwardBackward[100];

int StartPositionUpDown[100];

const int startBT = 9;

const int teachBT = 10;

const int stopBT = 10;

const int potForwardBackward = A3; //ใช้

const int potRotate_Grib = A4;

const int potUpDown_Grib = A5;

const int potUpDown = A2;//ใช้

const int potRotate = A1;

const int potGrab = A0;

//Max and Min values for servos ! Change them to meet your setup !

const int minGrab=180;

const int maxGrab=126;

const int minRotate=0;

const int maxRotate=100;

const int minUpDown=5;// ใช้

const int maxUpDown=100;// ใช้

const int minRotate_Grib=0;

const int maxRotate_Grib=100;
```

```
const int minUpDown_Grib=0;

const int maxUpDown_Grib=100;

const int minForwardBackward=160; //  ใช้

const int maxForwardBackward=80; //  ใช้

void setup() {

  Serial.begin(9600);

  pinMode(in1, OUTPUT);

  pinMode(in2, OUTPUT);

  pinMode(enA, OUTPUT);

  pinMode(teachBT, INPUT_PULLUP);

  pinMode(startBT, INPUT_PULLUP);

  SPI.begin();

  SPI.setDataMode (SPI_MODE1) ;

  SPI.setBitOrder(MSBFIRST); }

void loop() {

  enc();

  readInputs(); //อ่านค่า pot

  GoToAngle(readUpDown, e);}
```

```
void moveMotor()
{
  GoToAngle(readUpDown, e);
}
void GoToAngle(int target, int encode)
{ int current = encode;
  Serial.print("current = ");
  Serial.println(current);
  Serial.print("target = " );
  Serial.println(target);
  if(target>current&&target-current!=PERM_ERROR&&target-current>PERM_ERROR)
  { turnleft();
    Serial.println("turnleft");
}
  else if(target<current&&current-target!=PERM_ERROR&&current-
target>PERM_ERROR)
  {
    turnright();
    Serial.println("turnright");
```

```
  }
  else if(target==current||target-current==PERM_ERROR||current-
target==PERM_ERROR||0<current-target<PERM_ERROR||0<target-
current<PERM_ERROR)
  {
    Stop();
    delay(50);
  }
}
void SetTurnDirection(int dir)
{//  Serial.println("turn");
  turnDirection = dir;
  switch (turnDirection)
  {
    case 1: //turning Right
      digitalWrite(in1, HIGH);
      digitalWrite(in2, LOW);
      Turn();
      //Serial.println("turn_right");
```

```
      break;
     case 2: //turning Left
      digitalWrite(in1, LOW);
      digitalWrite(in2, HIGH);
      Turn();
      //Serial.println("turn_left");
      break;
  }
}
void turnright()
{
  digitalWrite(in1, HIGH);
      digitalWrite(in2, LOW);
      Turn();
}
void turnleft()
{
  digitalWrite(in1, LOW);
      digitalWrite(in2, HIGH);
      Turn();
```

```
}


void Turn()
{
  analogWrite(enA, turnSpeed);
}
void Stop()
{
  analogWrite(enA, 0);
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
 Serial.println("Stop");
}


void readInputs(){
  //Read potentiometers
readUpDown = analogRead(potUpDown); //
  readUpDown = map(readUpDown,0,1023,minUpDownAngle,maxUpDownAngle);
//บังคับ pot จำกัดค่ามุมอยู่แค่ระหว่างช่วง min-max
  readTeach = digitalRead(teachBT);
```

```
  readStart = digitalRead(startBT);
  readStop = digitalRead(stopBT);}
void savePosition(){
  teachUpDown[index] = readUpDown;
  teachForwardBackward[index] = readForwardBackward;
  teachRotate[index] = readRotate;
  teachGrab[index] = readGrab;
  teachRotate_Grib[index] = readRotate_Grib;
  teachUpDown_Grib[index] = readUpDown_Grib;
  index++;}
void enc()
{ digitalWrite(SS, LOW);
  result1 = SPI.transfer(0b00000000);
  result1 &= 0b00111111;
  result1 = result1 << 8;
  result2 = SPI.transfer(0b00000000);
  result = result1 | result2;
  digitalWrite(SS, HIGH);
  e = map(result, 0, 16383, 0, 360); //นำค่าจาก encoder มาแปลงเป็นองศา
  //Serial.println(e);}
```

ประวัติผู้เขียน

ชื่อ – ชื่อสกุล	ณัฐชนน ศุภอดิเรก
วัน เดือน ปี เกิด	16 กันยายน 2539
ที่อยู่ปัจจุบัน	บ้านเลขที่ 99/1068 ถนนราษฎร์พัฒนา22 แขวงสะพานสูง แขวงสะพานสูง จังหวัดกรุงเทพมหานคร 10240

ประวัติการศึกษา

พ.ศ. 2547-2553	สำเร็จการศึกษาระดับชั้นประถมจากโรงเรียนเอกบูรพาวิเทศศึกษา
พ.ศ. 2553-2556	สำเร็จการศึกษาระดับชั้นมัธยมจากโรงเรียนเตรียมอุดมศึกษาน้อมเกล้า
พ.ศ. 2556-2559	สำเร็จการศึกษาระดับชั้นมัธยมจากโรงเรียนเตรียมอุดมศึกษาน้อมเกล้า
พ.ศ. 2559-2562	หลักสูตรวิศวกรรมแมคคาทรอนิกส์ ภาควิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประวัติการฝึกงาน

พ.ศ. 2561	โตโยต้า มอเตอร์ ประเทศไทย จำกัด (โรงงานบ้านโพธิ์)
พ.ศ. 2561	บริษัท เด็นโซ่ (ประเทศไทย) จำกัด (โรงงานบางปะกง)