



รายงานสหกิจฉบับสมบูรณ์

ระบบควบคุมน้ำ

Water Supply Control System

นายอภิสิทธิ์ เต็มคุณานนท์

ภาควิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2562

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงการสหกิจศึกษา ระบบควบคุมน้ำ (Water Supply Control System)

ชื่อโครงการสหกิจศึกษา ระบบควบคุมน้ำ

ชื่อ-สกุลนักศึกษา นาย อภิสิตธิ์ เต็มคุณานนท์

คณะ วิศวกรรมศาสตร์

ภาควิชา วิศวกรรมไฟฟ้า

ชื่อ-สกุล อาจารย์นิเทศ ดร. วิวัฒน์ เกรยรวงศ์

ชื่อ-สกุล ผู้นิเทศงาน นาย ภิญญ ต่ออำนาจ

สถานประกอบการ บริษัท เคมีแมน จำกัด (สาขาโรงงานแก่งคอย จังหวัดสระบุรี)

บทคัดย่อ

รายงานสหกิจศึกษานี้ได้รับการสนับสนุนจาก บริษัท เคมีแมน จำกัด (มหาชน) แผนกซ่อมบำรุงไฟฟ้าและวัดคุม ที่ได้ให้โอกาสเข้าร่วมในการออกแบบระบบควบคุมน้ำ ในพื้นที่ของบริษัทโดยมีวัตถุประสงค์เพื่อเพิ่มประสิทธิภาพของระบบน้ำให้ได้ปริมาณน้ำที่ไว้ในกิจกรรมต่าง ๆ ของบริษัทมากกว่าเดิม ซึ่งโครงการนี้จะเป็นการประยุกต์ใช้การควบคุมทางอุตสาหกรรมในระบบอัตโนมัติ โดยจะใช้ตัวควบคุมเป็นไมโครคอนโทรลเลอร์ ARDUINO นอกจากนี้ยังสามารถควบคุม การทำงานได้ในโหมดการควบคุมด้วยมือ (Manual) โดยสั่งการทำงานผ่านปุ่ม ON-OFF ของตู้ควบคุมเครื่องจักรบริเวณหน้างาน

คำสำคัญ : ARDUINO

Cooperative Title : Water Supply Control System

Student intern name : Mr. Apisit Termkunanon


Faculty : Engineering

Department : Electrical Engineering

Advisor name : Dr.Wiwat Keyoonwong

Mentor name : Mr. Pin Toramnart

Company : Chememan Co.,Ltd (Kaeng khoi Saraburi)



ABSTRACT

This cooperative educational report is supported from Chememan Co, Ltd in Electrical Maintenance and Instrument Department, to give opportunity for cooperative student to design the Water Supply Control System in area of company, the objective to increase the efficiency of the water system to get more water used in the company's activities than before, This project is an application of industrial control in automation. By using the controller as a ARDUINO microcontroller, Moreover, the operator can order each machine operation in manual mode by push ON-OFF switch on local area of machine control cabinet.

Keyword : ARDUINO

กิตติกรรมประกาศ

โครงการสหกิจศึกษาเรื่องระบบควบคุมน้ำฉบับนี้ สำเร็จลุล่วงไปได้ด้วยการ สนับสนุนจากหลายฝ่ายที่ให้ คำปรึกษาและชี้แนะแนวทาง ทำให้โครงการสหกิจศึกษานี้บรรลุวัตถุประสงค์ไปได้ด้วยดี ผู้เขียนจึงขอกราบขอบพระคุณไว้ ณ โอกาสนี้

ผู้เขียนขอขอบพระคุณ ดร.วิวัฒน์ เกรยวรงค์ อาจารย์ปรึกษาโครงการสหกิจศึกษาที่คอย ให้ คำปรึกษา ชี้แนะและดูแล ตลอดระยะเวลาในการปฏิบัติสหกิจศึกษา รวมถึงช่วยตรวจทาน และแก้ไขข้อบกพร่องต่าง ๆ ทำให้รายงานสหกิจศึกษานี้สมบูรณ์

ขอขอบพระคุณบริษัท เคมีแมน จำกัด (มหาชน) ที่มอบโอกาสในการเข้ามาทำสหกิจศึกษา ประจำปีการศึกษา 2561 รวมถึง คุณ ภิญ ต่ออำนาจ ซึ่งเป็นผู้ดูแลที่คอยให้คำปรึกษา และชี้แนะแนวทาง ตลอดการทำโครงการ

อภิสิทธิ์ เต็มคุณานนท์

สารบัญ

หน้า

บทคัดย่อ.....	i
ABSTRACT.....	ii
กิตติกรรมประกาศ.....	iii
บทที่ 1 บทนำ.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขอบเขตของการทำงาน.....	2
1.4 ขั้นตอนการทำงาน.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	4
2.1 การควบคุมอัตโนมัติทางอุตสาหกรรม [1].....	4
2.1.1 การควบคุม.....	4
2.1.2 การควบคุมอัตโนมัติทางอุตสาหกรรม.....	4
2.1.3 การควบคุมลำดับ.....	4
2.2 มอเตอร์ไฟฟ้า (Motor).....	4
2.2.1 มอเตอร์ไฟฟ้ากระแสสลับชนิด 3 เฟส (A.C. Three phase Motor).....	5
2.2.2 วิธีการเริ่มเดินมอเตอร์.....	6
2.2.3 ประเภทของการควบคุมมอเตอร์.....	8
2.3 อุปกรณ์ไฟฟ้า.....	10
2.3.1 Circuit Breaker.....	10

2.3.2	ฟิวส์ (Fuse)	11
2.3.3	รีเลย์ (Relay)	11
2.3.4	แมกเนติกคอนแทคเตอร์ (Magnetic Contactor)	12
2.3.5	สวิตช์ปุ่มกด (Push Button Switch).....	13
2.3.6	ซีเล็คเตอร์สวิตช์ (Selector Switch).....	13
2.3.7	สวิตช์ซึ่งเพาเวอร์ซัพพาย (Switching Power Supply).....	14
2.3.8	โซลินอยด์วาว์ว (Solenoid Valve).....	15
2.4	ไมโครคอนโทรลเลอร์ (Micro Controller)	15
2.4.1	องค์ประกอบของไมโครคอนโทรลเลอร์ (Micro Controller Component).....	15
2.5	อาร์ดูโน้ (ARDUINO)	17
2.5.1	ส่วนประกอบของบอร์ด Arduino	17
2.5.2	ภาษาที่ใช้เขียนโปรแกรมใน Arduino.....	19
2.5.3	Arduino Mega Wifi ATmega2560+ESP8266	22
บทที่ 3 วิธีการดำเนินการ.....		24
3.1	ศึกษาการทำระบบน้ำจากระบบเดิมของโรงงาน และออกแบบระบบน้ำใหม่.....	24
3.1.1	สำรวจระบบน้ำแบบเก่า	24
3.1.2	ออกแบบระบบน้ำแบบใหม่	25
3.2	การออกแบบระบบควบคุม	26
3.2.2	เลือกอุปกรณ์ระบบไฟฟ้า.....	26
3.2.3	ออกแบบตู้ควบคุม.....	30
3.2.4	เขียนโปรแกรม Arduino.....	40
3.3.1	แผนผังการทำงาน (FLOW CHART)	41

3.3.2 เขียนโปรแกรมลงใน ARDUINO	42
บทที่ 4 ผลการดำเนินการ	48
4.1 แบบวงจรควบคุมระบบน้ำ	48
4.1.1 แบบวงจรควบคุมระบบของตู้ควบที่1	48
4.1.2 63แบบวงจรควบคุมระบบของตู้ควบที่2	51
4.1.3 แบบวงจรควบคุมระบบของตู้ควบที่3	54
4.2 โปรแกรมควบคุมArduino	57
4.2.1ทำงานทดสอบโปรแกรมที่อัฟโหลดลงบอร์ดArduino	57
บทที่ 5 สรุป ปัญหา ข้อเสนอแนะ	61
5.1 สรุป	61
5.2 ปัญหา และข้อเสนอแนะ	61
บรรณานุกรม	62
ประวัติผู้เขียน	63

สารบัญภาพ

ภาพที่ 2.1 มอเตอร์เหนี่ยวนำสามเฟสแบบกรงกระรอก	5
ภาพที่ 2.2 วงจรกำลังสตาร์ทแบบ DOL.....	6
ภาพที่ 2.3 กราฟเปรียบเทียบวิธีสตาร์ทแบบ DOL และ STAR-DELTA	7
ภาพที่ 2.4 Soft Start ABB รุ่น PSTX VFD ย่อมาจาก Variable Frequency Drives.....	7
ภาพที่ 2.5 Inverter ABB รุ่น ACS550.....	8
ภาพที่ 2.6 ฟังการควบคุมมอเตอร์ด้วยมือ	9
ภาพที่ 2.7 ฟังการควบคุมมอเตอร์กึ่งอัตโนมัติ	9
ภาพที่ 2.8 ฟังการควบคุมมอเตอร์อัตโนมัติ	10
ภาพที่ 2.9 Circuit Break Schneider รุ่น HDF26015	11
ภาพที่ 2.10 ฟิวส์ (Fuse)	11
ภาพที่ 2.11 รีเลย์ (Relay).....	12
ภาพที่ 2.11 โซลิดสเตทรีเลย์ (Solid state Relay).....	12
ภาพที่ 2.12 แมกเนติกคอนแทคเตอร์ (Magnetic Contactor)	13
ภาพที่ 2.13 สวิตช์ปุ่มกด (Push Button Switch).....	13
ภาพที่ 2.14 ซีเล็คเตอร์สวิตช์ (Selector Switch).....	14
ภาพที่ 2.15 สวิตช์ซึ่งเพาเวอร์ซัพพาย (Switching Power Supply).....	14
ภาพที่ 2.16 โซลินอยด์ยวาวว (Solenoid Valve)	15
ภาพที่ 2.17 องค์ประกอบของไมโครคอนโทรลเลอร์ (Micro Controller Component).....	16
ภาพที่ 2.18 ส่วนประกอบของอาร์ดูโน (ARDUINO Component).....	17
ภาพที่ 2.19 ส่วนประกอบของอาร์ดูโน (ARDUINO Component).....	18

ภาพที่ 2.20 ส่วนประกอบของอาร์ดูโน้ (ARDUINO Component)	18
ภาพที่ 2.21 ส่วนประกอบของอาร์ดูโน้ (ARDUINO Component)	19
ภาพที่ 2.22 Arduino Mega Wifi ATmega2560+ESP8266.....	23
ภาพที่ 3.2 ออกหน้างานเพื่อวัดหาความยาวขอท่อน้ำ.....	25
ภาพที่ 3.3 ระบบน้ำระบบใหม่ของโรงงาน	25
ภาพที่ 3.4 Submersible Pump.....	27
ภาพที่ 3.5 SELECTION CHART	27
ภาพที่ 3.6 Booster Pump & Control Box.....	28
ภาพที่ 3.7 สายคอนโทรล LIYCY HELUKABEL	28
ภาพที่ 3.8 ท่อร้อยสาย HDPE.....	29
ภาพที่ 3.9 สวิตช์ลุ่มกลอย(Float Switch)	29
ภาพที่ 3.10 สวิตช์แรงดัน(Pressure Switch).....	30
ภาพที่ 3.11 บริเวณที่ติดตั้งตู้คอนโทรล	30
ภาพที่ 3.12 Booster Pump & Control Box.....	31
ภาพที่ 3.13 ไฟสัญญาณ(Pilot Lamp) สีต่างๆ	32
ภาพที่ 3.14 มาตรฐานสีของไฟแจ้งเตือน.....	33
ภาพที่ 3.15 สวิตช์ปุ่มกด(Push Botton Switch).....	33
ภาพที่ 3.16 สวิตช์ดอกเห็ด(Emergency Stop)	34
ภาพที่ 3.17 สวิตช์ตัวเลือก (Selector Switch).....	34
ภาพที่ 3.18 สวิตช์ตัวเลือก (Selector Switch).....	35
ภาพที่ 3.19 โมดูลรีเลย์ 24 โวลต์ 8 chanal.....	36
ภาพที่ 3.20 โมดูลรีเลย์โวลิตสเตท 5โวลต์.....	36

ภาพที่ 3.21 เซอร์กิตเบรกเกอร์ แบบ 2 ขั้ว (ซ้าย) ละ 1 ขั้ว (ขวา).....	37
ภาพที่ 3.22 สวิตซ์ชิงเพาเวอร์ซัพพาย (Switching Powersupply).....	37
ภาพที่ 3.23 Cable Grand.....	38
ภาพที่ 3.24 Cable Grand.....	38
ภาพที่ 3.25 รางปีกนก (Omega rail).....	39
ภาพที่ 3.26 Cable Duct	39
ภาพที่ 3.27 สายคอนโทรล (Control Cable)	40
ภาพที่ 2.28 1 Arduino Mega Wifi ATmega2560+ESP8266.....	40
ภาพที่ 3.29 Interface ของ Program Arduino IDE	42
ภาพที่ 3.30 Interface ของ Program Arduino IDE	42
ภาพที่ 3.31 Interface ของ Program Arduino IDE	43
ภาพที่ 3.32 Interface ของ Program Arduino IDE	43
ภาพที่ 3.33 Interface ของ Program Arduino IDE	44
ภาพที่ 3.34 ภาพการกำหนดตัวแปรในโปรแกรม Arduino IDE.....	44
ภาพที่ 3.35 การกำหนด OUTPUT INPUT ในโปรแกรม Arduino IDE.....	45
ภาพที่ 3.36 กำหนดให้ OUTPUT มีสถานะเป็น HIGH.....	45
ภาพที่ 3.37 การทำงานแบบวนซ้ำ (LOOP)	46
ภาพที่ 3.38 การตรวจสอบความถูกต้องของไวยากรณ์ (COMPILE).....	46
ภาพที่ 3.39 การอัปโหลดโปรแกรมลงไปในบอร์ด Arduino.....	47
ภาพที่ 4.1 แบบวงจรควบคุมตู้ที่ 1 (1)	48
ภาพที่ 4.2 แบบวงจรควบคุมตู้ที่ 1 (2)	49
ภาพที่ 4.3 แบบวงจรควบคุมตู้ที่ 1 (3)	49

ภาพที่ 4.4 แบบวงจรควบคุมตู้ที่1 (4)	50
ภาพที่ 4.5 แบบวงจรควบคุมตู้ที่1 (5)	50
ภาพที่ 4.6 แบบวงจรควบคุมตู้ที่1 (6)	51
ภาพที่ 4.7 แบบวงจรควบคุมตู้ที่2 (1)	51
ภาพที่ 4.8 แบบวงจรควบคุมตู้ที่2 (2)	52
ภาพที่ 4.9 แบบวงจรควบคุมตู้ที่2 (3)	52
ภาพที่ 4.10 แบบวงจรควบคุมตู้ที่2 (4)	53
ภาพที่ 4.11 แบบวงจรควบคุมตู้ที่2 (5)	53
ภาพที่ 4.12 แบบวงจรควบคุมตู้ที่2 (6)	54
ภาพที่ 4.13 แบบวงจรควบคุมตู้ที่3 (1)	54
ภาพที่ 4.14 แบบวงจรควบคุมตู้ที่3 (2)	55
ภาพที่ 4.15 แบบวงจรควบคุมตู้ที่3 (3)	55
ภาพที่ 4.16 แบบวงจรควบคุมตู้ที่3 (4)	56
ภาพที่ 4.17 แบบวงจรควบคุมตู้ที่3 (5)	56
ภาพที่ 4.18 แบบวงจรควบคุมตู้ที่3 (6)	57
ภาพที่ 4.19 การทดสอบการทำงานของ Arduino	57

สารบัญตาราง

ตารางที่ 3.1 การเปรียบเทียบระหว่าง ARDUINO กับ PLC.....	26
ตารางที่ 3.2 ตารางบอกสีของไฟแจ้งเตือนที่ใช้.....	32
ตารางที่ 4.1 การทดสอบการทำงานของARDUINOตัวที่1.....	58
ตารางที่ 4.2 การทดสอบการทำงานของARDUINOตัวที่2.....	59
ตารางที่ 4.3 การทดสอบการทำงานของARDUINOตัวที่3.....	60



บทที่ 1 บทนำ

1.1 ความเป็นมาและความสำคัญ

บริษัท เคมีแมน จำกัด (มหาชน) ก่อตั้งขึ้นเมื่อปี พ.ศ. 2546 โดยกลุ่มผู้บริหารมีอาชีพที่มีประสบการณ์ ที่หลากหลายและเป็นที่ยอมรับในด้านการพัฒนาอุตสาหกรรม เพื่อดำเนินธุรกิจผลิตแร่และเคมีอุตสาหกรรม โดยแปรสภาพจากแร่ธรรมชาติสู่ผลิตภัณฑ์แร่เคมีพื้นฐานและเคมีสังเคราะห์ ในปัจจุบันกิจการของบริษัทครอบคลุมตลาดในทวีปเอเชีย ออสเตรเลียและแอฟริกา บริษัท เคมีแมน จำกัด (มหาชน) เป็นที่รู้จักกันดีในฐานะผู้ผลิต ปูนโลม์ชั้นแนวหน้าของทวีปเอเชียและกำลังขยายทางธุรกิจที่กว้างขวางและทางลึก ทั้งในประเทศและต่างประเทศ โดยมีเป้าหมายเพิ่มกำลังการผลิตปูนโลม์เป็น 1 ล้านตันต่อปี ซึ่งจะทำให้บริษัท เคมีแมน จำกัด (มหาชน) ก้าวสู่การเป็นผู้ผลิตปูนโลม์ 1 ใน 10 อันดับแรกของโลก ภายในปี พ.ศ. 2563 ในปัจจุบันบริษัท เคมีแมน จำกัด (มหาชน) มีโครงการที่จะเพิ่มกำลังการผลิต ให้มากขึ้นเพื่อให้สอดคล้องกับเป้าหมายของบริษัท จึงมีความต้องการน้ำใช้ในกิจกรรมต่างๆภายในโรงงานมากขึ้นจึงต้องจัดทำระบบน้ำใหม่ โดยหน้าที่ที่นักศึกษารับผิดชอบคือ ออกแบบระบบไฟและชุดควบคุมเพื่อใช้ในการควบคุมการทำงานของมอเตอร์ของเครื่องสูบน้ำ รวมไปถึงการเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ ARDUINO ควบคุมการทำงานของเครื่องจักร โดยให้เครื่องจักรแต่ละตัวสามารถทำงานร่วมกันเป็นระบบได้ และเมื่อมีเหตุขัดข้องสามารถให้เจ้าหน้าที่ซ่อมบำรุง (Maintenance) ให้เข้ามาแก้ไขความผิดพลาดได้อย่างรวดเร็วและแม่นยำ ทั้งนี้ผู้ปฏิบัติงานสามารถสั่งการทำงานของ เครื่องจักรในโหมดการควบคุมด้วยมือ (Manual) ผ่านตู้ควบคุมเครื่องจักรบริเวณหน้างาน อย่างไรก็ตามในส่วนที่นักศึกษาได้รับมอบหมายให้รับผิดชอบนั้นจะเป็นการออกแบบระบบไฟฟ้า และระบบควบคุมแต่เนื่องจากการผิดพลาดเรื่องระยะเวลาในการจัดส่งอุปกรณ์ทำให้ดำเนินการติดตั้งไม่ทัน ดังนั้นในรายงานฉบับนี้จะกล่าวถึงการออกแบบไฟฟ้าและโปรแกรมที่ใช้ในการควบคุมระบบน้ำ

1.2 วัตถุประสงค์

1.2.1 การปฏิบัติงานตามที่ได้รับมอบหมาย

- 1) ทำการบำรุงรักษาเครื่องจักรประจำปี
- 2) ตรวจสอบระบบไฟฟ้า และเครื่องจักรไฟฟ้า

1.2.2 การปฏิบัติงานโครงการ

1) ออกแบบวงจรควบคุมของโครงการระบบน้ำ ในส่วนของการควบคุมด้วยระบบควบคุมอัตโนมัติ โดยใช้ ไมโครคอนโทรลเลอร์

2) เขียนโปรแกรมเพื่อนควบคุมไมโครคอนโทรลเลอร์ โดยใช้ Arduino เป็นตัวควบคุม ในโครงการระบบน้ำระบบใหม่

1.3 ขอบเขตของการทำงาน

- 1) ออกแบบการวางอุปกรณ์ในตู้ควบคุมมอเตอร์และออกแบบตู้ที่ใช้ในการควบคุมเครื่องจักร
- 2) จัดทำแบบไฟฟ้าในการติดตั้ง (Wiring Diagram) ของอุปกรณ์ แสดงการเชื่อมต่อสายระหว่างอุปกรณ์ เพื่อที่จะ ทำให้ช่างไฟฟ้ามีความเข้าใจง่ายขึ้นเมื่อทำการเดินสายไฟของอุปกรณ์
- 3) ตรวจสอบแบบวงจรไฟฟ้าให้มีความเหมาะสมที่จะนำไปใช้ในการควบคุมมอเตอร์ของเครื่องจักร
- 4) เขียนโปรแกรมควบคุมกระบวนการทำงานของระบบควบคุมน้ำ
- 5) ดำเนินการติดตั้งอุปกรณ์ภายในตู้ควบคุมและติดตั้งที่หน้างาน
- 6) ทำสอบใช้งานติดตามผลและดำเนินการแก้ไขข้อผิดพลาด

1.4 วิธีการดำเนินงาน

- 1) ศึกษากระบวนการระบบเดิมของโรงงาน และออกสำรวจพื้นที่จริงพร้อมทำการวัดระยะของท่อส่งน้ำ
- 2) ศึกษาการเลือกใช้ขนาดของอุปกรณ์ เช่น เซอร์กิตเบรกเกอร์ แมกเนติกส์คอนแทคเตอร์ รีเลย์และชุดการเริ่มเดินมอเตอร์ เพื่อควบคุมการทำงานของมอเตอร์ของเครื่องจักร และศึกษาการเลือกขนาดของสายไฟให้ มีขนาดที่จะรองรับกำลังไฟฟ้าจากมอเตอร์ของเครื่องจักรได้
- 3) ศึกษาการเลือกขนาดกำลังของมอเตอร์เพื่อให้เพียงพอต่อการสูบ และส่งน้ำ
- 4) ออกแบบการวางอุปกรณ์ในตู้ควบคุม รวมถึงการเลือกใช้สายไฟให้มีขนาด เหมาะสมกับขนาดกำลังไฟฟ้าของมอเตอร์เครื่องจักรในแต่ละประเภท และเลือกอุปกรณ์ต่าง ๆ ให้เหมาะสม
- 5) จัดทำแบบไฟฟ้าที่ใช้ในการควบคุมมอเตอร์
- 6) ศึกษาการเขียนโปรแกรม Arduino เพื่อควบคุมการทำงานของมอเตอร์

7) ศึกษาการทำตู้ควบคุมมอเตอร์ และติดตั้งตู้ควบคุมบริเวณหน้างาน

1.5 แผนการดำเนินงาน

หัวข้องาน	เดือนที่ 1	เดือนที่ 2	เดือนที่ 3	เดือนที่ 4
1. ศึกษากระบวนการระบบเก่าในโรงงานแก่งคอย	■			
2. ออกแบบระบบน้ำระบบใหม่		■		
3. ศึกษาการทำงานของอุปกรณ์พร้อมเลือกใช้			■	
4. สั่งซื้ออุปกรณ์				■
5. เขียนโปรแกรม				■
6. ทดสอบอุปกรณ์				■

1.6 ประโยชน์ที่คาดว่าจะได้รับ

- 1) สามารถออกแบบการวางและเลือกใช้อุปกรณ์ในตู้ควบคุมได้
- 2) สามารถเลือกใช้อุปกรณ์ให้เหมาะสมกับขนาดกำลังของมอเตอร์
- 3) สามารถออกแบบระบบควบคุมการทำงานของระบบการโดยใช้ไมโครคอนโทรลเลอร์ได้
- 4) ได้เห็นรายละเอียดและเข้าใจขั้นตอนการทำงาน of โครงการต่างๆ ภายในบริษัทตั้งแต่เริ่มต้นพร้อมทั้งได้รับประสบการณ์ในการทำงานจริง ที่สามารถนำไปประยุกต์ใช้กับการทำงานในอนาคต
- 6) มีความรับผิดชอบต่อนหน้าที่ที่ได้รับมอบหมาย และมีกำคิดวิเคราะห์อย่างเป็นระบบ

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

2.1 การควบคุมอัตโนมัติทางอุตสาหกรรม [1]

2.1.1 การควบคุม เป็นวิธีการที่ทำให้เครื่องจักรทำงานได้ถูกต้องตามเงื่อนไขที่กำหนด ซึ่งการควบคุมในงานอุตสาหกรรมทำ เพื่อควบคุมเครื่องจักรให้ทำงานในกระบวนการผลิตตรงตามจุดประสงค์ของโรงงานอุตสาหกรรม

2.1.2 การควบคุมอัตโนมัติทางอุตสาหกรรม เป็นการควบคุมเครื่องจักรที่ใช้ในกระบวนการผลิตให้สามารถทำงานแทนมนุษย์ได้ โดยมีความผิดพลาด ในการผลิตน้อยลง เวลาการผลิตเร็วขึ้น ลดการใช้แรงงาน และความสิ้นเปลืองของพลังงานลง

2.1.3 การควบคุมลำดับ เป็นการควบคุมขั้นตอนการทำงานของเครื่องจักรการผลิตในงานอุตสาหกรรมให้ทำงานสอดคล้องกันตาม ขั้นตอนการผลิตของโรงงานอุตสาหกรรม โดยมีอุปกรณ์ตรวจวัด (Sensors) ทำหน้าที่ตรวจสอบการทำงานของ เครื่องจักร และอุปกรณ์ควบคุม (Control Devices) ที่ทำหน้าที่ควบคุมเครื่องจักรการผลิตให้ได้ผลผลิตตามต้องการ

2.2 มอเตอร์ไฟฟ้า (Motor) [2]

เป็นอุปกรณ์ไฟฟ้าที่เปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกล มอเตอร์ที่ใช้งานในปัจจุบัน แต่ละชนิดก็จะมีคุณสมบัติที่แตกต่างออกไปต้องการความเร็วรอบ หรือกำลังงานที่ต่างกันไป ซึ่งมอเตอร์แต่ละชนิด จะแบ่งได้เป็น 2 ชนิดตามลักษณะการใช้งาน กระแสไฟฟ้า

1) มอเตอร์ไฟฟ้ากระแสสลับ (Alternating Current Motor) หรือเรียกว่าเอ.ซี.มอเตอร์ (A.C. MOTOR) การแบ่งชนิดของมอเตอร์ไฟฟ้าสลับแบ่งออกเป็น 3 ชนิดได้แก่

1.1) มอเตอร์ไฟฟ้ากระแสสลับชนิด 1 เฟส หรือเรียกว่าซิงเกิลเฟสมอเตอร์ (A.C. Sing Phase) จะใช้กับแรงดันไฟฟ้า 220 โวลต์มีสายไฟ เข้า 2 สาย มีแรงม้าไม่สูง ส่วนใหญ่ตามบ้านเรือน

- สปลิตเฟส มอเตอร์ (Split-Phase motor)
- คาปาซิเตอร์มอเตอร์ (Capacitor motor)
- รีพัลชั่นมอเตอร์ (Repulsion-type motor)

- ยูนิเวอร์แซลมอเตอร์ (Universal motor)

- เซ็ดเดดโพล มอเตอร์ (Shaded-pole motor)

1.2) มอเตอร์ไฟฟ้าสลับชนิด 2 เฟสหรือเรียกว่าทูเฟสมอเตอร์ (A.C. Two phase Motor)

1.3) มอเตอร์ไฟฟ้ากระแสสลับชนิด 3 เฟสหรือเรียกว่าทรีเฟสมอเตอร์ (A.C. Three phase Motor) เป็น มอเตอร์ที่ใช้ในงานอุตสาหกรรมต้องใช้ระบบไฟฟ้า 3 เฟส ใช้แรงดัน 400 โวลต์ มีสายไฟเข้ามอเตอร์ 3 สาย

2) มอเตอร์ไฟฟ้ากระแสตรง (Direct Current Motor) หรือเรียกว่าดี.ซี มอเตอร์ (D.C. MOTOR) การแบ่งชนิดของมอเตอร์ไฟฟ้ากระแสตรงแบ่งออกได้ดังนี้ มอเตอร์ไฟฟ้ากระแสตรงแบ่งออกเป็น 3 ชนิด ได้แก่

2.1) มอเตอร์แบบอนุกรมหรือเรียกว่าซีรีส์มอเตอร์ (Series Motor)

2.2) มอเตอร์แบบขนานหรือเรียกว่าชันทมอเตอร์ (Shunt Motor)

2.3) มอเตอร์ไฟฟ้าแบบผสมหรือเรียกว่าคอมพาวด์มอเตอร์ (Compound Motor)

2.2.1 มอเตอร์ไฟฟ้ากระแสสลับชนิด 3 เฟส (A.C. Three phase Motor)

มอเตอร์เหนี่ยวนำ 3 เฟส เป็นมอเตอร์ที่นิยมใช้งานกันทั่วไปในโรงงานอุตสาหกรรม โดยเฉพาะมอเตอร์เหนี่ยวนำ 3 เฟสชนิดที่มีโรเตอร์แบบกรงกระรอกมีข้อดี คือ ไม่มีแปรงถ่านทำให้การสูญเสียเนื่องจากความฝืดมีค่าน้อย มีตัวประกอบกำลังสูง การบำรุงรักษาต่ำ การเริ่มต้นทำได้ไม่ยาก ความเร็วรอบค่อนข้างคงที่ สร้างง่าย ทนทาน ราคาถูก และมีประสิทธิภาพสูง แต่มีข้อเสีย ได้แก่ การเปลี่ยนแปลงความเร็วรอบของมอเตอร์ทำได้ยาก ปัจจุบันได้มีการพัฒนาชุดควบคุมอินเวอร์เตอร์ใช้สำหรับปรับความเร็วรอบของมอเตอร์และเป็นที่ยอมรับกันอย่างแพร่หลาย



ภาพที่ 2.1 มอเตอร์เหนี่ยวนำสามเฟสแบบกรงกระรอก [3]

2.2.2 วิธีการเริ่มเดินมอเตอร์ [4]

การควบคุมมอเตอร์ไฟฟ้าแบบ AC 3 Phase Induction Motor หรือ Asynchronous Motor นั้นสามารถควบคุมให้มอเตอร์เริ่มทำงาน Start Motor และควบคุมให้มอเตอร์หยุดทำงาน Stop Motor ได้หลากหลายวิธี แต่สำหรับในประเทศไทยที่ใช้ระบบไฟฟ้าแบบ 3 เฟส 400 โวลต์ 50Hz จะนิยมใช้การควบคุม มอเตอร์ 2 แบบ คือ

1) Mechanical Starters (DOL)

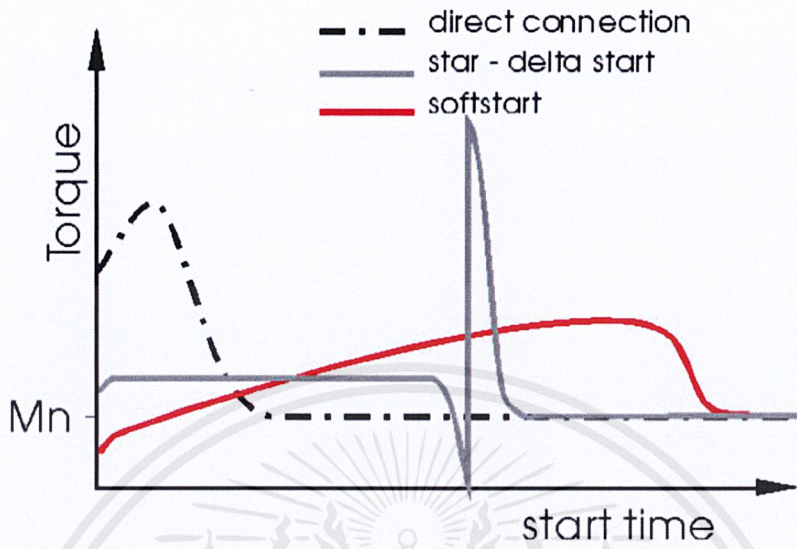
เป็นการสตาร์ทมอเตอร์แบบ Direct Online ด้วยแมกเนติกคอนแทคเตอร์ โดย จะสตาร์ทด้วยแรงดันเต็มพิกัด (Full-Voltage Starting) วิธีการสตาร์ทมอเตอร์แบบนี้เป็นที่นิยมกันมาก ใช้ สำหรับมอเตอร์ที่มีขนาดเล็ก ซึ่งมอเตอร์จะถูกต่อผ่านอุปกรณ์สตาร์ทแล้วต่อเข้ากับสายไฟกำลังโดยตรง ทำให้มอเตอร์สตาร์ทด้วยแรงดันเท่ากับสายจ่ายแรงดันทันทีทันทีใด และกระแสขณะสตาร์ทสูงถึงประมาณ 6 - 8 เท่าของกระแสพิกัด อาจทำให้เกิดอันตรายต่อมอเตอร์ หรือวงจรไฟฟ้าอื่น ที่ต่อร่วมสายจ่ายกำลังมอเตอร์ได้



ภาพที่ 2.2 วงจรกำลังสตาร์ทแบบ DOL

2) STAR-DELTA Timer Start

ในภาคอุตสาหกรรม และ ในเครื่องจักรประเภทต่างๆ มอเตอร์ เป็นอุปกรณ์ไฟฟ้าที่เป็นส่วนสำคัญและมีอัตราในการใช้พลังงานไฟฟ้าสูงมาก โดยส่วนมากในโรงงานอุตสาหกรรมนิยมใช้งานมอเตอร์แบบ 3 เฟส ซึ่งเป็นมอเตอร์ที่กินกระแสสูงในขณะสตาร์ท (Start Motor) โรงงานอุตสาหกรรมต่างจึงมีความต้องการที่จะลดกระแสไฟฟ้าในขณะสตาร์ท การต่อมอเตอร์แบบ สตาร์เดลต้า (Star Delta) จึงเป็นวิธีหนึ่งที่จะช่วยให้เรา สามารถลดกระแสไฟฟ้าในขณะสตาร์ทได้ การใช้ **Star-Delta Timer** สามารถช่วยให้ง่ายมากขึ้น สำหรับการต่อวงจรไฟฟ้าในการควบคุม มอเตอร์สตาร์ เดลต้า (Star Delta)



ภาพที่ 2.3 กราฟเปรียบเทียบวิธีสตาร์ทแบบ DOL และ STAR-DELTA

3) Electronic Starters (Soft Starters, VFDs)

ซอฟสตาร์ท (Soft Start) คือ อุปกรณ์ที่นำมาใช้กับการสตาร์ทของมอเตอร์ เพื่อลดการกระชากของกระแสไฟฟ้าในช่วงของการสตาร์ทของมอเตอร์ ทำให้มอเตอร์และกลไกขับเคลื่อนต่างๆ มีอายุการใช้งานที่ยาวนานขึ้น และยังส่งผลต่อการอนุรักษ์พลังงานไฟฟ้า ทำให้ค่าใช้จ่ายลดตามลงด้วยเช่นกัน



ภาพที่ 2.4 Soft Start ABB รุ่น PSTX [5]

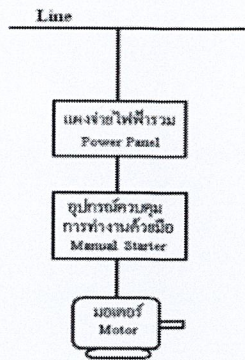
VFD ย่อมาจาก Variable Frequency Drives หรือไม่ก็อาจจะเรียกกันว่า VSD (Variable Speed Drives) หรือไม่ก็ อินเวอร์เตอร์ เป็นอุปกรณ์ที่ใช้สำหรับควบคุมแรงบิดและความเร็วรอบของมอเตอร์ และแน่นอน- นอนสามารถที่จะใช้สำหรับการสตาร์ท และสตอปมอเตอร์ได้อีกด้วย



ภาพที่ 2.5 Inverter ABB รุ่น ACS550 [6]

2.2.3 ประเภทของการควบคุมมอเตอร์ [7]

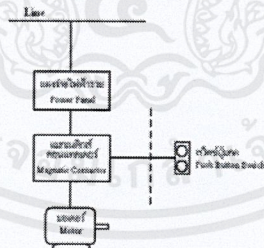
1) การควบคุมด้วยมือ (Manual Control) การควบคุมด้วยมือ เป็นการสั่งงานให้อุปกรณ์ควบคุมทำงานโดยใช้ผู้ปฏิบัติงานควบคุมให้ระบบกลไกทางกลทำงานซึ่งการสั่งงานให้ระบบกลไกทำงานนี้โดยส่วนมากจะใช้คนเป็นผู้สั่งงานแทบทั้งสิ้น ซึ่งมอเตอร์จะถูกควบคุมจากการสั่งงานด้วยมือโดยการควบคุมผ่านอุปกรณ์ต่าง ๆ เช่น พุชบอททอนสวิทช์ (Push Button Switch) เซฟตี้ สวิทช์ (Safety Switch) ดรัมสวิทช์ (Drum Switch) ตัวควบคุมแบบหน้างาน (Face Plate Control) เป็นต้น



ภาพที่ 2.6 แผงการควบคุมมอเตอร์ด้วยมือ

2) การควบคุมกึ่งอัตโนมัติ (Semi-Automatic Control)

โดยการใช้สวิตช์ปุ่มกด (Push Button) ที่สามารถควบคุมระยะไกล (Remote Control) ได้ ซึ่งมักจะต่อร่วมกับสวิตช์แม่เหล็ก (Magnetic Switch) ที่ใช้จ่ายกระแสจำนวนมากๆ ให้กับมอเตอร์แทนสวิตช์ธรรมดา ซึ่ง สวิตช์แม่เหล็กนี้อาศัยผลการทำงานของแม่เหล็กไฟฟ้า วงจรการควบคุมมอเตอร์กึ่งอัตโนมัตินี้ต้องอาศัยคนคอย กดสวิตช์จ่ายไฟให้กับสวิตช์แม่เหล็กสวิตช์แม่เหล็กจะดูดให้หน้าสัมผัสมาแตะกันและจ่ายไฟให้กับมอเตอร์ และถ้าต้องการหยุดมอเตอร์ก็จะต้องอาศัยคนคอยกดสวิตช์ปุ่มกดอีกเช่นเดิม จึงเรียกรูปแบบนี้ว่าการควบคุมกึ่งอัตโนมัติ

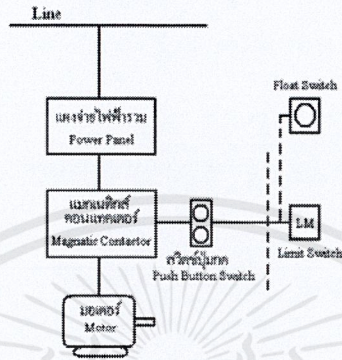


ภาพที่ 2.7 แผงการควบคุมมอเตอร์กึ่งอัตโนมัติ

3) การควบคุมอัตโนมัติ (Automatic Control)

การควบคุมแบบนี้จะอาศัยอุปกรณ์ชิ้น ๑ (Pilot Device) คอยตรวจจับการเปลี่ยนแปลงของสิ่งต่าง ๆ เช่น สวิตช์-ลูกลอย ทาหน้าที่ตรวจวัดระดับน้ำ ภาในถัง คอยสั่งให้มอเตอร์ปั๊ม ทางานเมื่อน้ำ าทมดถึง และสั่งให้มอเตอร์ หยุดเมื่อน้ำ าทเต็มถึง, สวิตช์ความดัน (Pressure Switch) ทาหน้าที่ตรวจจับความดันลมเพื่อสั่งให้

ป้อนลมท างาน, เทอร์โมสแตท ท านหน้าที่ตัดต่อวงจรไฟฟ้าตามอุณหภูมิสูงหรือต่ำ เป็นต้น วงจรการควบคุมมอเตอร์แบบนี้เพียงแต่ ใช้คนกดปุ่มเริ่มเดินมอเตอร์ในครั้งแรกเท่านั้น ต่อไปวงจรก็จะท างานเองโดยอัตโนมัติตลอดเวลา



ภาพที่ 2.8 ผังการควบคุมมอเตอร์อัตโนมัติ

2.3 อุปกรณ์ไฟฟ้า [8]

2.3.1 Circuit Breaker

เซอร์กิตเบรกเกอร์หรือเบรกเกอร์ คือ สวิตช์ไฟฟ้าอัตโนมัติที่ออกแบบมาเพื่อป้องกันวงจรไฟฟ้าจากความเสียหายที่เกิดจากกระแสไฟฟ้าส่วนเกิน โดยทั่วไปเกิดจากโหลดเกินหรือไฟฟ้าลัดวงจร การทำงานของมันคือตัด กระแสไฟฟ้าหลังจากตรวจพบความผิดปกติในวงจรไฟฟ้าถือว่าเป็นอุปกรณ์ที่ใช้ป้องกันกระแสเกิน หรือลัดวงจร เช่นเดียวกับฟิวส์ แต่จะแตกต่างกันตรงที่เมื่อตัดวงจรแล้วสามารถที่จะปิดหรือต่อวงจรได้ทันทีหลังจากแก้ปัญหา แล้ว เบรกเกอร์มีหลายแบบทั้งเบรกเกอร์ขนาดเล็กที่ใช้ป้องกันสำหรับวงจรที่มีกระแสไฟฟ้าต่ำ หรือพวกเครื่องใช้ไฟฟ้าในครัวเรือน จนถึงสวิตช์ขนาดใหญ่ที่ออกแบบมาเพื่อป้องกันวงจรไฟฟ้าแรงสูงที่จ่ายไฟให้ตัวเมือง



ภาพที่ 2.9 Circuit Break Schneider รุ่น HDF26015

2.3.2 ฟิวส์ (Fuse)

เป็นอุปกรณ์ป้องกันวงจรไฟฟ้าจากการที่มีกระแสไหลผ่านวงจรมากเกินไป (Overload Current) หรือเกิดไฟฟ้าลัดวงจร (Short Circuit Current) เมื่อมีกระแสที่มากกว่ากระแสที่ฟิวส์ทนได้ (Current Rating) ลักษณะการทำงานคือเมื่อมีกระแสไฟฟ้าไหลผ่านฟิวส์จะเกิดการเปลี่ยนพลังงานไฟฟ้าเป็นพลังงานความร้อน ให้กับฟิวส์เล็กน้อย แต่ถ้ากระแสไฟฟ้าที่ไหลผ่านฟิวส์มีค่ามากเกินไป (Overload Current) จะทำให้พลังงาน ความร้อนที่เกิดขึ้นนั้นมีค่ามากจนฟิวส์หลอมละลายได้เนื่องจากฟิวส์นั้นทำจากโลหะที่มีจุดหลอมเหลวต่ำจึงทำให้วงจรขาดได้ง่ายและเกิดการตัดกระแสไฟออกจากวงจรไฟฟ้าทันที เพื่อเป็นการป้องกันความเสียหายที่จะเกิดขึ้น ซึ่งโดยปกติแล้วกระแสเกินพิกัด (Overload Current) นั้นเกิดเมื่อมีการดึงกระแสที่มากเกินไปจากโหลด ส่วน กระแสลัดวงจร (Short Circuit Current) เกิดจากการที่กระแสเคลื่อนที่ผ่านทางลัดที่อาจเกิดจากการแตะกัน ของสายไฟหรือมีตัวนำไฟฟ้าเชื่อมต่อการลัดวงจรจาก L-N หรือ L-L

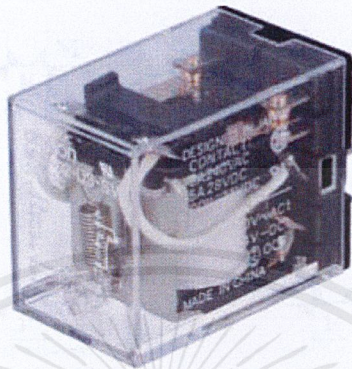


ภาพที่ 2.10 ฟิวส์ (Fuse)

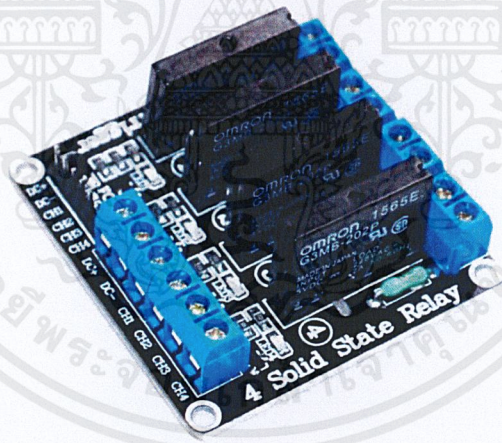
2.3.3 รีเลย์ (Relay)

คืออุปกรณ์อิเล็กทรอนิกส์ที่ทำหน้าที่เป็นสวิตช์ตัด-ต่อวงจร โดยใช้แม่เหล็กไฟฟ้า และการที่จะให้รีเลย์ทำงานก็ต้องจ่ายไฟให้อุปกรณ์ เมื่อรีเลย์ได้รับการจ่ายไฟ จะทำให้หน้าสัมผัสติดกัน กลายเป็นวงจรปิด

และตรงข้ามทันทีที่ไม่ได้จ่ายไฟให้รีเลย์ ก็จะกลายเป็นวงจรเปิด ไฟที่เราใช้ป้อนให้กับตัวรีเลย์ก็จะเป็นไฟ
ที่มาจากเพาเวอร์ของอุปกรณ์ไฟฟ้า ดังนั้นทันทีที่เปิดเครื่องก็จะทำให้รีเลย์ทำงาน



ภาพที่ 2.11 รีเลย์ (Relay)

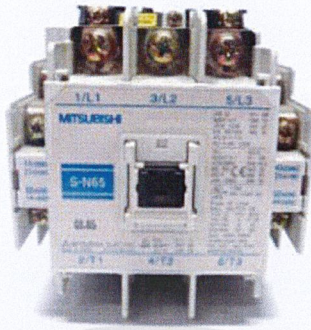


ภาพที่ 2.11 โซลิดสเตตรีเลย์ (Solid state Relay)

2.3.4 แมกเนติกคอนแทคเตอร์ (Magnetic Contactor)

แมกเนติกคอนแทคเตอร์ คืออุปกรณ์สวิตช์ตัดต่อวงจรไฟฟ้า เพื่อการเปิด-ปิด ของหน้าสัมผัส (Contact) ทำงานโดยอาศัยอำนาจแม่เหล็กไฟฟ้าช่วยในการเปิด-ปิดหน้าสัมผัส ในการตัดต่อวงจรไฟฟ้า เช่น เปิด-ปิด การทำงานของวงจรควบคุมมอเตอร์ นิยมใช้ในวงจรของระบบแอร์, ระบบควบคุมมอเตอร์ หรือใช้ในการ

ควบคุมเครื่องจักรต่างๆ โดยแมกเนติกคอนแทคเตอร์นั้น จะมีส่วนประกอบหลักที่สำคัญต่อการทำงาน ได้แก่ แกนเหล็ก (Core),ขดลวด (Coil),หน้าสัมผัส (Contact) และสปริง (Spring)



ภาพที่ 2.12 แมกเนติกคอนแทคเตอร์ (Magnetic Contactor)

2.3.5 สวิตช์ปุ่มกด (Push Button Switch)

Push Button Switch หรือที่เรียกกันว่าสวิตช์ปุ่มกด เป็นอุปกรณ์ทางไฟฟ้าซึ่งทำหน้าที่ตัดและต่อวงจรทางไฟฟ้าและ ใช้ในการควบคุมการทำงานของมอเตอร์ หรือการทำงานของเครื่องจักรต่างๆ เป็นเหมือนอุปกรณ์ พื้นฐาน ใช้ได้กับอุตสาหกรรมทั่วไป มีทั้งแบบมีไฟ และทึบแสง



ภาพที่ 2.13 สวิตช์ปุ่มกด (Push Button Switch)

2.3.6 ซีเล็คเตอร์สวิตช์ (Selector Switch)

เป็นอุปกรณ์ที่ใช้ควบคุมวงจรไฟฟ้าภาคคอลโทรล เพื่อควบคุมทิศทางของกระแสไฟฟ้าให้ตามทิศทางที่ต้องการ หรือตัดกระแสไฟไม่ให้ไหลผ่านวงจรได้ตามที่ต้องการ เป็นสวิตช์ที่ใช้งานกันมากในงานที่ต้องควบคุม การทำงานด้วยมือ โดยการบิดให้คอนแทค ที่อยู่ภายในเปลี่ยนสถานะปกติปิด (NC) หรือปกติเปิด (NO) โดย ซีเล็คเตอร์ สวิตช์ ทั่วไปจะมี 2 ประเภท คือ แบบ สวิตช์ 2 ทางและสวิตช์ 3 ทาง

- 1) สวิตช์ 2 ทาง หรือ 2 Position : เหมาะสำหรับงานออกคำสั่งการทำงานของเครื่องจักร 1 คำสั่ง เช่น ใช้ในการเปิด-ปิด เป็นต้น นิยมใช้ในการควบคุม เปิดหรือปิดปั๊มน้ำ
- 2) สวิตช์ 3 ทาง หรือ 3 Position : เหมาะสำหรับใช้ควบคุมเครื่องจักรที่มากกว่า 1 คำสั่ง เช่น ตำแหน่ง 1-0-2 เมื่อสวิตช์ไปที่ตำแหน่ง 1 จะทำให้มอเตอร์หมุนไปทางทิศตามเข็มนาฬิกา และเมื่อบิดมาที่ตำแหน่ง 0 มอเตอร์จะหยุดทำงาน และเมื่อบิดไปที่ตำแหน่ง 2 มอเตอร์จะหมุนไปทางทิศทวนเข็มนาฬิกา เป็นต้น ตัวอย่างเช่น มอเตอร์ที่ใช้ในการสูบน้ำ



ภาพที่ 2.14 ซีเล็กเตอร์สวิตช์ (Selector Switch)

2.3.7 สวิตซ์ซิ่งเพาเวอร์ซัพพาย (Switching Power Supply)

สวิตซ์ซิ่งเพาเวอร์ซัพพาย (Switching Power Supply) เป็นแหล่งจ่ายไฟตรงคงค่าแรงดันแบบหนึ่ง และสามารถเปลี่ยนแรงดันไฟจากไปสลับโวลต์สูง ให้เป็นแรงดันไฟตรงค่าต่ำ เพื่อใช้ในงานอิเล็กทรอนิกส์ และไปไฟเลี้ยงให้กันไมโครคอนโทรลเลอร์อีกด้วย



ภาพที่ 2.15 สวิตซ์ซิ่งเพาเวอร์ซัพพาย (Switching Power Supply)

2.3.8 โซลินอยด์วาล์ว (Solenoid Valve)

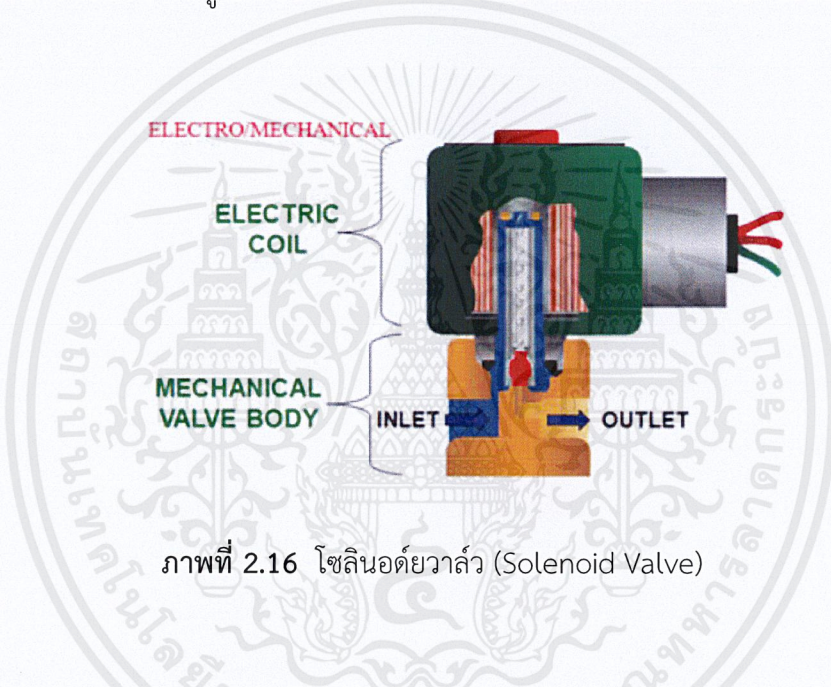
อุปกรณ์ทางอิเล็กทรอนิกส์ แมกเนติก ที่ใช้ควบคุมปริมาณของ ของไหลที่ไหลผ่านท่อ โดยการเปิดหรือปิดที่รู orifices ของตัววาล์ว

โครงสร้างของโซลินอยด์วาล์ว

โซลินอยด์วาล์วเป็นการรวมกันของ 2 รูปแบบการทำงาน คือ

SOLENOID - (Electro-magnetic) coil จะเป็นตัวทำหน้าที่สร้างสนามแม่เหล็กและเหนี่ยวนำให้ plunger เคลื่อนที่ขึ้นลง

VALVE – ตัววาล์วจะมีรู orifice ที่มี disc คอยปิดและเปิดให้ของไหลไหลผ่านวาล์ว



ภาพที่ 2.16 โซลินอยด์วาล์ว (Solenoid Valve)

2.4 ไมโครคอนโทรลเลอร์ (Micro Controller) [9]

คือ อุปกรณ์ควบคุมขนาดเล็ก ซึ่งบรรจุความสามารถที่คล้ายคลึงกับระบบคอมพิวเตอร์ โดยในไมโครคอนโทรลเลอร์ได้รวมเอาซีพียู, หน่วยความจำ และพอร์ต ซึ่งเป็นส่วนประกอบหลักสำคัญของระบบคอมพิวเตอร์เข้าไว้ด้วยกัน โดยทำการบรรจุเข้าไว้ในตัวถังเดียวกัน

2.4.1 องค์ประกอบของไมโครคอนโทรลเลอร์ (Micro Controller Component)

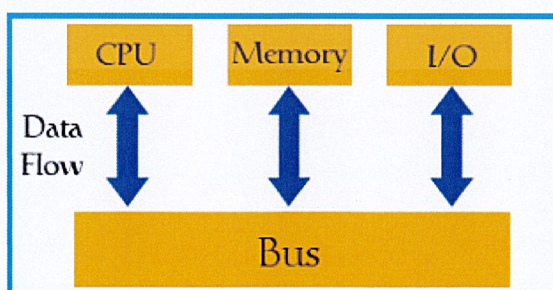
- 1) หน่วยประมวลผลกลางหรือซีพียู (CPU : Central Processing Unit)

2) หน่วยความจำ (Memory) สามารถแบ่งออกเป็น 2 ส่วน คือ หน่วยความจำที่มีไว้สำหรับเก็บโปรแกรมหลัก (Program Memory) เปรียบเสมือนฮาร์ดดิสก์ของเครื่องคอมพิวเตอร์ตั้งโต๊ะ คือ ข้อมูลใดๆ ที่ถูกเก็บไว้ในนี้จะไม่สูญหายไปแม้ไม่มีไฟเลี้ยง อีกส่วนหนึ่งคือหน่วยความจำข้อมูล (Data Memory) ใช้เป็นเหมือนกับกระดาษทดในการคำนวณของซีพียู และเป็นที่พักข้อมูลชั่วคราวขณะทำงาน แต่หากไม่มีไฟเลี้ยง ข้อมูลก็จะหายไปคล้ายกับหน่วยความจำแรม (RAM) ในเครื่องคอมพิวเตอร์ทุกๆ ไป แต่สำหรับไมโครคอนโทรลเลอร์สมัยใหม่ หน่วยความจำข้อมูลจะมีทั้งที่เป็นหน่วยความจำแรม ซึ่งข้อมูลจะหายไปเมื่อไม่มีไฟเลี้ยง และเป็นอีอีพรอม (EEPROM : Electrically Erasable Programmable Read-Only Memory) ซึ่งสามารถเก็บข้อมูลได้แม้ไม่มีไฟเลี้ยงก็ตาม

3) ส่วนติดต่อกับอุปกรณ์ภายนอก หรือพอร์ต (Port) มี 2 ลักษณะคือ พอร์ตอินพุต (Input Port) และพอร์ตส่งสัญญาณหรือพอร์ตเอาต์พุต (Output Port) ส่วนนี้จะใช้ในการเชื่อมต่อกับอุปกรณ์ภายนอก ถือว่าเป็นส่วนที่สำคัญมาก ใช้ร่วมกันระหว่างพอร์ตอินพุต เพื่อรับสัญญาณ อาจจะใช้การกดสวิตช์ เพื่อนำไปประมวลผลและส่งไปพอร์ตเอาต์พุต เพื่อแสดงผลเช่น การติดสว่างของหลอดไฟ เป็นต้น

4) ช่องทางเดินของสัญญาณ หรือบัส (BUS) คือเส้นทางการแลกเปลี่ยนสัญญาณข้อมูลระหว่างซีพียู หน่วยความจำและพอร์ต เป็นลักษณะของสายสัญญาณ จำนวนมากอยู่ภายในตัวไมโครคอนโทรลเลอร์ โดยแบ่งเป็นบัสข้อมูล (Data Bus) บัสแอดเดรส (Address Bus) และบัสควบคุม (Control Bus)

5) วงจรกำเนิดสัญญาณนาฬิกา เป็นองค์ประกอบที่สำคัญมากอีกส่วนหนึ่ง เนื่องจากการทำงานที่เกิดขึ้นในตัวไมโครคอนโทรลเลอร์ จะขึ้นอยู่กับกำหนัดจังหวะ หากสัญญาณนาฬิกามีความถี่สูง จังหวะการทำงานก็จะสามารถทำได้ถี่ขึ้นส่งผลให้ไมโครคอนโทรลเลอร์ตัวนั้น มีความเร็วในการประมวลผลสูงตามไปด้วย



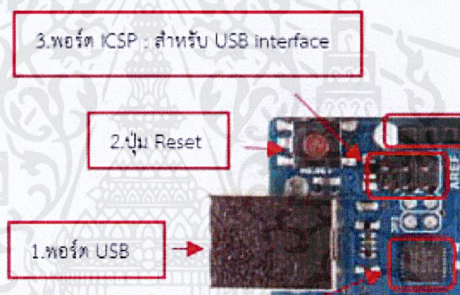
ภาพที่ 2.17 องค์ประกอบของไมโครคอนโทรลเลอร์ (Micro Controller Component)

2.5 อาร์ดูโน้ (ARDUINO) [10]

เป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัว บอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ด หรือ โปรแกรมต่อได้อีกด้วย

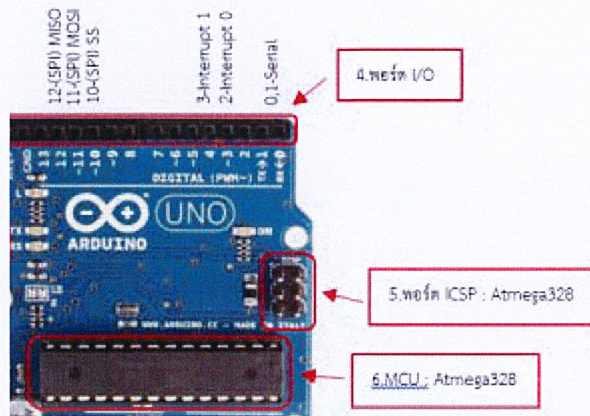
2.5.1 ส่วนประกอบของบอร์ด Arduino

- 1.USBPort: ใช้สำหรับต่อกับ Computer เพื่ออัปโหลดโปรแกรมเข้า MCU และจ่ายไฟให้กับบอร์ด
- 2.Reset Button: เป็นปุ่ม Reset ใช้กดเมื่อต้องการให้ MCU เริ่มการทำงานใหม่
- 3.ICSP Port ของ Atmega16U2 เป็นพอร์ตที่ใช้โปรแกรม Visual Com port บน Atmega16U2



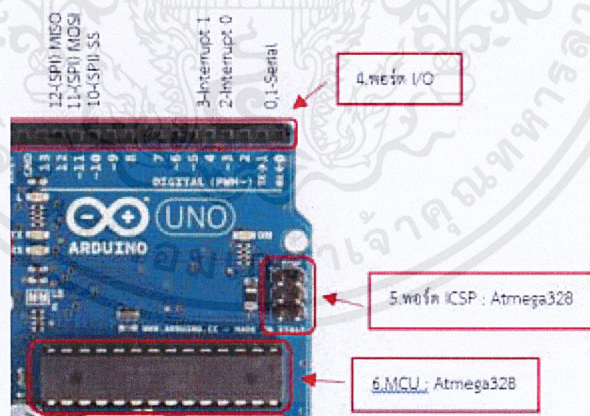
ภาพที่ 2.18 ส่วนประกอบของอาร์ดูโน้ (ARDUINO Component)

4. I/O Port: Digital I/O ตั้งแต่ขา D0 ถึง D13 นอกจากนี้ บาง Pin จะทำหน้าที่อื่นๆ เพิ่มเติมด้วย เช่น Pin0,1 เป็นขา Tx,Rx Serial, Pin3,5,6,9,10 และ 11 เป็นขา PWM
- 5.ICSP Port: Atmega328 เป็นพอร์ตที่ใช้โปรแกรม Bootloader
6. MCU: Atmega328 เป็น MCU ที่ใช้บนบอร์ด Arduino

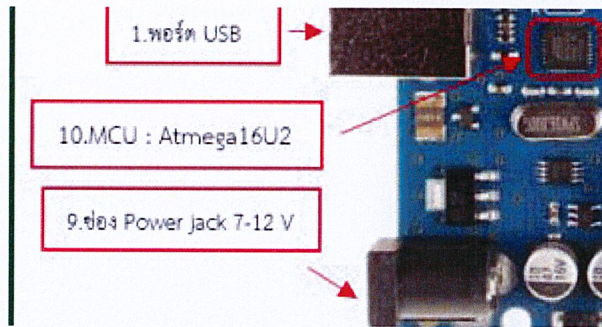


ภาพที่ 2.19 ส่วนประกอบของอาร์ดูโน้ (ARDUINO Component)

7. I/O Port: นอกจากจะเป็น Digital I/O แล้ว ยังเปลี่ยนเป็น ช่องรับสัญญาณอนาล็อก ตั้งแต่ขา A0-A5
8. Power Port: ไฟเลี้ยงของบอร์ดเมื่อต้องการจ่ายไฟให้กับวงจรภายนอก ประกอบด้วยขา ไฟเลี้ยง +3.3 V, +5V, GND, V_{in}
9. Power Jack: รับไฟจาก Adapter โดยที่แรงดันอยู่ระหว่าง 7-12 V
10. MCU ของ Atmega16U2 เป็น MCU ที่ทำหน้าที่เป็น USB to Serial โดย Atmega328 จะ ติดต่อกับ Computer ผ่าน Atmega16U2



ภาพที่ 2.20 ส่วนประกอบของอาร์ดูโน้ (ARDUINO Component)



ภาพที่ 2.21 ส่วนประกอบของอาร์ดูโน้ (ARDUINO Component)

2.5.2 ภาษาที่ใช้เขียนโปรแกรมใน Arduino [11]

โครงสร้างโปรแกรมภาษา C บน Arduino จะมีลักษณะแบบเดียวกับ C ทั่วไป แต่สำหรับท่านที่ยังไม่เคยเรียนรู้การเขียนโปรแกรมภาษาใดๆมาก่อน ท่านต้องทำความเข้าใจในเรื่องต่างๆดังนี้

1) ปรีโปรเซสเซอร์ไดเรกทีฟ (Preprocessor directives)

โดยปกติแล้วเกือบทุกโปรแกรมต้องมี โดยส่วนนี้จะเป็นส่วนที่คอมไพเลอร์จะมีการประมวลผล และทำตามคำสั่งก่อนที่จะมีการคอมไพล์โปรแกรม ซึ่งจะเริ่มต้นด้วยเครื่องหมายไดเรกทีฟ (directive) หรือเครื่องหมายสี่เหลี่ยม # แล้วจึงตามด้วยชื่อคำสั่งที่ต้องการเรียกใช้ หรือกำหนด โดยปกติแล้วส่วนนี้จะอยู่ในส่วนบนสุด หรือส่วนหัวของโปรแกรม และต้องอยู่นอกฟังก์ชันหลักใดๆก็ตาม

#include เป็นคำสั่งที่ใช้อ้างอิงไฟล์ภายนอก เพื่อเรียกใช้ฟังก์ชัน หรือตัวแปรที่มีการสร้างหรือกำหนดไว้ในไฟล์นั้น รูปแบบการใช้งานคือ

```
#include <ชื่อไฟล์.h>
```

ตัวอย่างเช่น

```
#include <Wire.h>
```

```
#include <Time.h>
```

จากตัวอย่าง จะเห็นว่าได้มีการอ้างอิงไฟล์ Wire.h และไฟล์ Time.h ซึ่งเป็นไลบรารีพื้นฐานที่มีอยู่ใน Arduino ทำให้เราสามารถใช้งานฟังก์ชันเกี่ยวกับเวลาที่ไลบรารี Time มีการสร้างไว้ให้ใช้งานได้ การอ้างอิงไฟล์จากภายใน หรือการอ้างอิงไฟล์ไลบรารีที่มีอยู่แล้วใน Arduino หรือเป็นไลบรารีที่เราเพิ่มเข้าไปเอง จะใช้เครื่องหมาย <> ในการคร่อมชื่อไฟล์ไว้ เพื่อให้โปรแกรมคอมไพเลอร์เข้าใจว่าควรไปหาไฟล์เหล่านี้จากในโพลเดอร์ไลบรารี แต่หากต้องการอ้างอิงไฟล์ที่อยู่ในโพลเดอร์โปร

เจค จะต้องใช้เครื่องหมาย "" ครอบแทน ซึ่งคอมไพเลอร์จะวิ่งไปหาไฟล์นี้โดยอ้างอิงจากไฟล์โปรแกรมที่คอมไพเลอร์อยู่

เช่น

```
#include "myFunction.h"
```

จากตัวอย่างด้านบน คอมไพเลอร์จะวิ่งไปหาไฟล์ myFunction.h ภายในโฟลเดอร์โปรเจคทันที หากไม่พบก็จะแจ้งเป็นข้อผิดพลาดออกมา

#define เป็นคำสั่งที่ใช้ในการแทนข้อความที่กำหนดไว้ ด้วยข้อความที่กำหนดไว้ ซึ่งการใช้คำสั่งนี้ ข้อดีคือไม่มีการอ้างอิงกับตัวโปรแกรมเลย

รูปแบบ

```
#define NAME VALUE
```

ตัวอย่างเช่น

```
#define LEDPIN 13
```

จากตัวอย่าง ไม่ว่าคำว่า LEDPIN จะอยู่ส่วนใดของโค้ดโปรแกรมก็ตาม คอมไพเลอร์จะแทนคำว่า LEDPIN ด้วยเลข 13 แทน ซึ่งข้อดีคือเราไม่ต้องสร้างเป็นตัวแปรขึ้นมาเพื่อเปลืองพื้นที่แรม และยังช่วยให้โปรแกรมทำงานเร็วขึ้นอีกด้วยเพราะซีพียูไม่ต้องไปขอข้อมูลมาจากแรมหลายๆทอด

2) ส่วนของการกำหนดค่า (Global declarations)

ส่วนนี้จะเป็นส่วนที่ใช้ในการกำหนดชนิดตัวแปรแบบนอกฟังก์ชัน หรือประกาศฟังก์ชัน เพื่อให้ฟังก์ชันที่ประกาศสามารถกำหนด หรือเรียกใช้ได้จากทุกส่วนของโปรแกรม

เช่น

```
int pin = 13;
```

```
void blink(void) ;
```

3) ฟังก์ชัน setup() และฟังก์ชัน loop()

ฟังก์ชัน setup() และฟังก์ชัน loop() เป็นคำสั่งที่ถูกบังคับให้ต้องมีในทุกโปรแกรม โดยฟังก์ชัน setup() จะเป็นฟังก์ชันแรกที่ถูกเรียกใช้ นิยมใช้กำหนดค่า หรือเริ่มต้นใช้งานไลบรารีต่างๆ เช่น ในฟังก์ชัน setup() จะมีคำสั่ง pinMode() เพื่อกำหนดให้ขาใดๆก็ตามเป็นดิจิตอลอินพุต หรือเอาต์พุต ส่วนฟังก์ชัน loop() จะเป็นฟังก์ชันที่ทำงานหลังจากฟังก์ชัน setup() ได้ทำงานเสร็จสิ้นไปแล้ว และมีการวนรอบแบบไม่รู้จบ เมื่อฟังก์ชัน loop() งานครบตามคำสั่งแล้ว ฟังก์ชัน loop() ก็จะถูกเรียกขึ้นมาใช้อีก

ตัวอย่าง

```
int pin = 13;
void setup() {
  pinMode(pin, OUTPUT);
}
void loop() {
  digitalWrite(pin, HIGH);
  delay(1000);
  digitalWrite(pin, LOW);
  delay(1000);
}
```

จากตัวอย่าง จะเห็นว่ามี การประกาศตัวแปรแบบนอกฟังก์ชัน ทำให้สามารถกำหนดหรือเรียกใช้ จากในฟังก์ชันใดๆก็ตามได้ ในฟังก์ชัน setup() ได้มีการกำหนดให้ขาที่ 13 เป็นดิจิตอลเอาต์พุต และในฟังก์ชัน loop() มีการกำหนดให้พอร์ต 13 มีลอจิกเป็น 1 และใช้ฟังก์ชัน delay() ในการ หน่วงเวลา 1 วินาที แล้วจึงกำหนดให้พอร์ต 13 มีสถานะลอจิกเป็น 0 แล้วจึงหน่วงเวลา 1 วินาที จบฟังก์ชัน loop() และจะเริ่มทำฟังก์ชัน loop() ใหม่ ผลที่ได้คือไฟกระพริบบนบอร์ด

ArduinoUnoในพอร์ตที่13ทำงานแบบไม่รู้จบ

ในทุกๆการทำงานของฟังก์ชัน จะต้องเริ่มด้วยการกำหนดค่าที่ส่งกลับ ตามด้วยชื่อฟังก์ชัน แล้ว ตามด้วยเครื่องหมายปีกกาเปิด { และจบด้วยเครื่องหมายปีกกาปิด ภายในฟังก์ชัน หากจะเรียก ฟังก์ชันใช้งานย่อยใดๆ จะต้องมีการเรียกชื่อฟังก์ชัน ; ต่อท้ายเสมอ

* การกำหนดชนิดค่าที่ส่งกลับเป็น void หมายถึงไม่มีการส่งค่ากลับ แต่สามารถใช้คำสั่ง return; ตรงๆได้ เพื่อให้จบการทำงานของฟังก์ชันก่อนจะไปถึงบรรทัดสุดท้ายของฟังก์ชัน

4. การสร้างฟังก์ชัน และการใช้งานฟังก์ชัน (Users-defined function)

ในการสร้างฟังก์ชันขึ้นมา คำสั่งต่างๆที่อยู่ภายในฟังก์ชัน ต้องอยู่ภายใต้เครื่องหมายปีกกาเปิด { และปีกกาปิด } เท่านั้น ภายใต้เครื่องหมาย {} เราสามารถนำฟังก์ชันหรือคำสั่งใดๆก็ได้มาใส่ไว้ แต่จะต้องคั่นแต่ละคำสั่งด้วยเครื่องหมายเซมิโคลอน ; โดยจะนำคำสั่งทั้งหมดไว้บรรทัดเดียวกัน เลย หรือแยกบรรทัดกันก็ได้เพื่อความสวยงามของโค้ด (ไม่มีผลกับขนาดของโปรแกรมหลัง คอมไพล์)

ตัวอย่าง

```

void Mode(int pin) {
    pinMode(pin, OUTPUT);
}

void setup() {
    Mode(13);
}

```

5. ส่วนอธิบายโปรแกรม (Program comments)

ส่วนอธิบายโปรแกรม หรือการคอมเมนต์โปรแกรมเป็นส่วนที่สำคัญอย่างมากที่จะช่วยให้ผู้ที่ไม่ได้เขียนโปรแกรม หรือเป็นผู้เขียนโปรแกรมเข้าใจโปรแกรมได้ง่ายขึ้นโดยอ่านจากคอมเมนต์ แทนการทำความเข้าใจโปรแกรมโดยอ่านแต่ละฟังก์ชัน ส่วนอธิบายโปรแกรม หรือส่วนคอมเมนต์นี้ จะไม่มีผลใดๆกับขนาดของโปรแกรมหลังคอมไพล์ เนื่องจากส่วนนี้จะถูกตัดทิ้งทั้งหมดเนื่องจากไม่ได้ถูกนำไปใช้งาน มีผลเพียงแค่ว่าไฟล์โค้ดโปรแกรมจะใหญ่ขึ้นมา หากมีการคอมเมนต์โค้ดเยอะๆ แต่ขนาดก็จะเพิ่มขึ้นตามตัวอักษร ดังนั้นการคอมเมนต์โค้ดจึงไม่คิดพื้นที่มากนัก แต่ผู้เขียนแนะนำให้คอมเมนต์โค้ดให้สั้น และกระชับ เพื่อให้เกิดความรวดเร็วในการทำความเข้าใจ และไม่ยาวจนต้องเลื่อนสกรีนบาร์ไปทางขวาเพื่ออ่านคอมเมนต์เพิ่มเติมอีก

การคอมเมนต์โค้ดมีอยู่ 2 รูปแบบ คือเปิดด้วย /* และปิดด้วย */ เป็นการคอมเมนต์โค้ดแบบข้ามบรรทัด คือตราบใดที่ยังไม่มี */ตรงส่วนนั้นจะเป็นคอมเมนต์ทั้งหมด เช่น

```

/*
This code by IOXhop.com
17/5/2558
*/

```

```

void setup() { ... }

```

และแบบที่ 2 เป็นการคอมเมนต์บรรทัดเดียว คือเปิดด้วยเครื่องหมาย // และปิดด้วยการขึ้นบรรทัดใหม่ เช่น

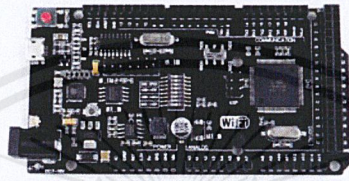
```

void setup() {
    pinMode(13, OUTPUT); // Set pin 13 to output
}

```

2.5.3 Arduino Mega Wifi ATmega2560+ESP8266

เป็นบอร์ด Arduino ที่ออกแบบมาสำหรับงานที่ต้องใช้ I/O มากกว่า Arduino Uno R3 เช่น งานที่ต้องการรับสัญญาณจาก Sensor หรือควบคุมมอเตอร์ Servo หลายๆ ตัว ทำให้ Pin I/O ของบอร์ด Arduino Uno R3 ไม่สามารถรองรับได้ ทั้งนี้บอร์ด Mega 2560 R3 ยังมีความหน่วยความจำแบบ Flash มากกว่า Arduino Uno R3 ทำให้สามารถเขียนโค้ดโปรแกรมเข้าไปได้มากกว่า ในความเร็วของ MCU ที่เท่ากัน



ภาพที่ 2.22 Arduino Mega Wifi ATmega2560+ESP8266

บทที่ 3

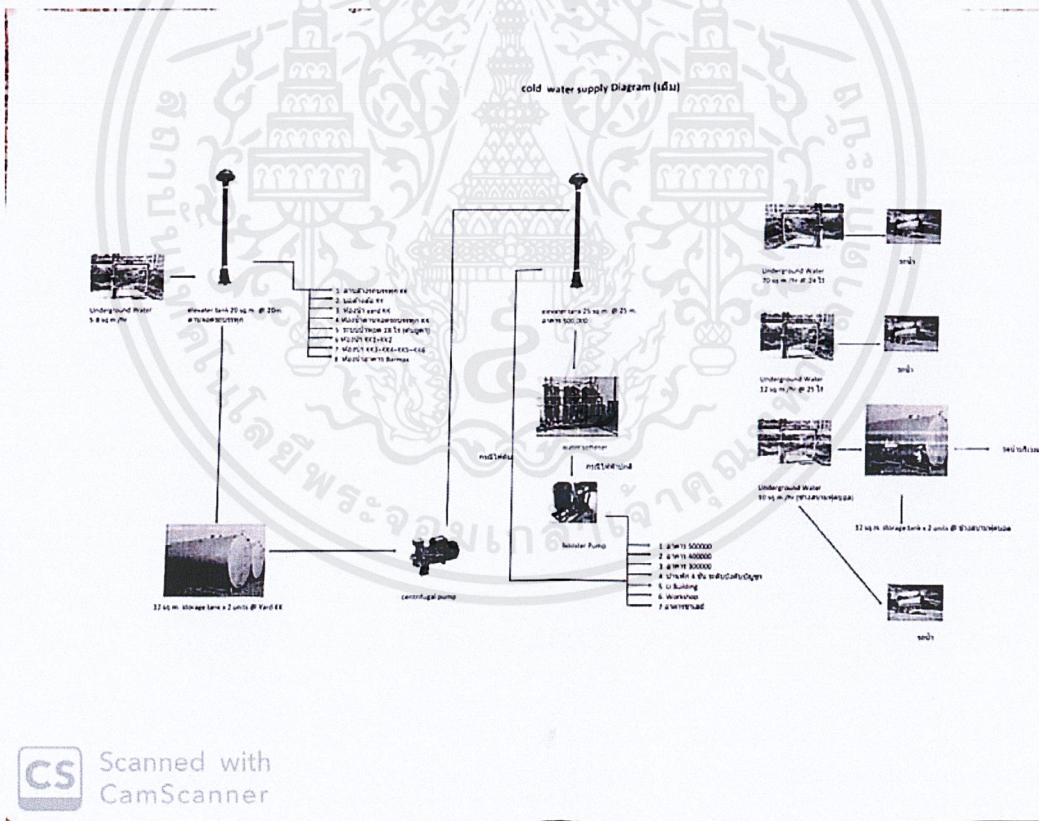
วิธีการดำเนินงาน

3.1 ศึกษาการทำระบบน้ำจากระบบเดิมของโรงงาน และออกแบบระบบน้ำใหม่

ทำการศึกษาระบบน้ำเดิมของโรงงาน และร่วมประชุมกับแผนกอาคารและสถานที่ของโรงงานเพื่อหาข้อมูลมาออกแบบระบบน้ำระบบใหม่ ซึ่งจากเดิมโรงงานใช้น้ำจากแทงค์แชมเปญู ที่บรรจุน้ำได้ปริมาตร 2500 ลิตรจำนวน2แทงค์ และได้มีการใช้รถขนน้ำเพื่อนำไปใช้บริเวณต่างๆ ต่อมาโรงงานต้องการที่จะเพิ่มแทงค์บรรจุน้ำที่สามารถบรรจุน้ำได้ 7500 ลิตร จำนวน4แทงค์ และลดการใช้รถขนน้ำเพื่อที่จะประหยัดน้ำมัน

3.1.1 สำรจระบบน้ำแบบเก่า

ลงพื้นที่จริงและสำรจอุปกรณ์ต่างๆของระบบน้ำระบบเดิมของโรงงาน



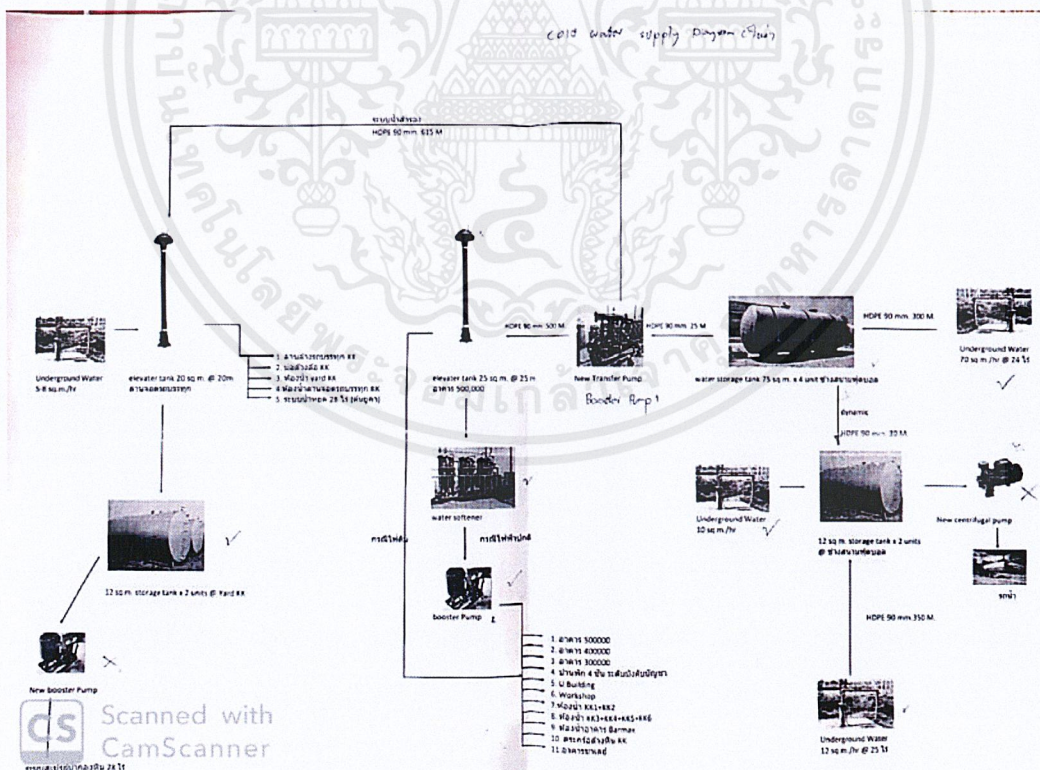
ภาพที่3.1 ระบบน้ำระบบเก่าของโรงงาน

3.1.2 ออกแบบระบบน้ำแบบใหม่

สอบถามความต้องการของบริษัท และลงพื้นที่หาความยาวของท่อหน้าและท่อร้อยสายไฟ เพื่อออกแบบระบบน้ำระบบใหม่



ภาพที่ 3.2 ออกหน้างานเพื่อวัดหาความยาวของท่อหน้า



ภาพที่ 3.3 ระบบน้ำระบบใหม่ของโรงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
25
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบระบบควบคุม

3.2.1 เลือกใช้คอนโทรลเลอร์

จากการที่ได้เปรียบเทียบคอนโทรลเลอร์ทั้ง 2ชนิดระหว่าง PLC (Programable Logic Control) และ Arduino จากการเปรียบเทียบความคุ้มค่าในแต่ละด้านแล้ว ได้เลือก Arduino เพราะ มีราคาถูกแล้ว สามารถใช้ควบคุมมอเตอร์ และ โซลินอยวาล์วได้ไม่ต่างจากPLC

	PLC	ARDUINO
1.การใช้งาน	✓	✓
2.การซ่อมบำรุง	✓	✓
3.ความทนทาน	✓	
4.ความน่าเชื่อถือ	✓	
5.ราคา		✓

ตารางที่ 3.1 การเปรียบเทียบระหว่างPLC กับ ARDUINO

3.2.2 เลือกอุปกรณ์ระบบไฟฟ้า

เริ่มจากการกำหนดเงื่อนไขตามที่ทางบริษัทต้องการ และในส่วนนี้จะใช้โปรแกรม AutoCAD ในการออกแบบระบบควบคุม โดยจะทำการออกแบบเป็น wiring diagramเพื่อใช้ในการติดตั้งอุปกรณ์ในตัวคอนโทรลเงื่อนไขที่ทางบริษัทกำหนดมาก็คือ ต้องการที่จะเพิ่มปริมาณการเก็บน้ำโดยจะเพิ่มถังเก็บน้ำขนาด7500ลิตร จำนวน4ถัง และยกเลิกการใช้รถน้ำเพื่อเป็นการลดต้นทุนด้านค่าน้ำมัน โดยคอนโทรลเลอร์ที่เลือกใช้คือ Arduinoจึงได้ทำการศึกษาอุปกรณ์ไฟฟ้าที่ต้องใช้ และได้ทำการออกแบบ

3.2.1.1เลือกมอเตอร์

ในระบบน้ำจะมีมอเตอร์ที่ใช้อยู่2ประเภท

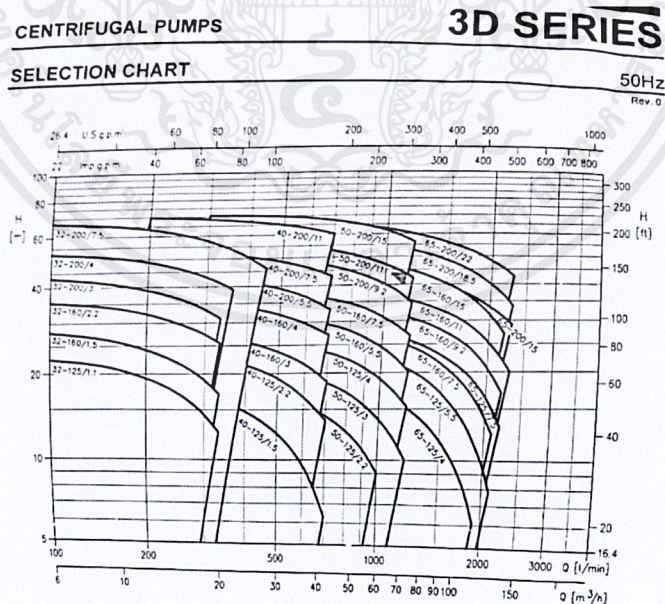
1.Submersible Pump หรือ บั้มแช่ จะใช้สูบน้ำจากบาดาล เนื่องจากทางโรงงานมีอยู่แล้วจึงไม่ต้องทำการสั่งซื้อใหม่



ภาพที่ 3.4 Submersible Pump

2. Booster Pump

Booster Pump หรือ ปั๊มเสริมแรงดัน คือระบบปั๊มน้ำที่ทำหน้าที่เพิ่มและรักษาแรงดันน้ำในระบบให้มีแรงดันที่สม่ำเสมอ มีหลักการทำงานคือ เมื่อมีการเปิดน้ำใช้จากระบบ ระดับแรงดันจะค่อยๆ ลดลงจนถึงค่าที่ตั้งไว้ สวิตช์แรงดัน (Pressure switch) จะสั่งให้ปั๊มน้ำทำการสูบน้ำเพื่อให้แรงดันในท่อค่อยๆ เพิ่มขึ้นถึงค่าที่กำหนดไว้ สวิตช์แรงดันก็จะสั่งให้ปั๊มหยุดทำงาน มีวิธีเลือกขนาดของปั๊ม โดยใช้วิธี เลือกจากกราฟ SELECTION CHART



โดยบริเวณใช้งานมีความสูงประมาณ 50 เมตร และต้องอัตราการไหล 600 ลิตร ต่อ นาที หรือ 50 คิว ต่อ ชั่วโมง

จากกราฟ จะได้ปั๊มขนาด 50-200/11 (11kW)



ภาพที่ 3.6 Booster Pump & Control Box

3) เลือกขนาดความยาวสายและขนาดความยาวของท่อร้อยสาย

ในการสั่งซื้อจะเผื่อความยาวไปอีก 20 เปอร์เซ็นต์ จากที่วัดได้จากหน้างาน โดยจะเลือกใช้สายคอนโทรลแบบ LIYCY 7 แกน ขนาด 1.5 ตารางมิลลิเมตร ที่เลือกแบบ 7 แกน เพราะว่าเพียงพอต่อเอาท์พุท และเนื่องจากทางบริษัทมีความต้องการให้เดินสายไฟแบบร้อยท่อฝังดิน จึงได้เลือกท่อร้อยสายไฟแบบ HDPE เนื่องจากมีความทนทานสูง



ภาพที่ 3.7 สายคอนโทรล LIYCY HELUKABEL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
28
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.8 ท่อร้อยสาย HDPE

4) สวิตช์ลู่กลอย(Float Switch)

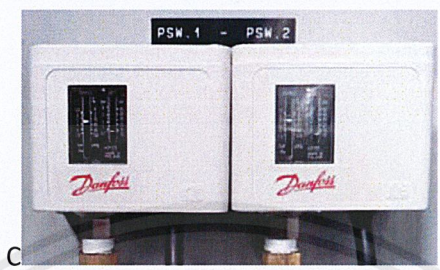
ใช้เป็นอุปกรณ์ในการควบคุมระดับน้ำในระดับสูงสุดและต่ำสุด เมื่อระดับน้ำถึงจุดที่ตั้งไว้หน้าสัมผัสก็จะเปลี่ยนสถานะ หน้าสัมผัสมีทั้งแบบปกติเปิด(NO) และแบบปกติปิด(NC)



ภาพที่ 3.9 สวิตช์ลู่กลอย(Float Switch)

5) สวิตช์แรงดัน (Pressure Switch)

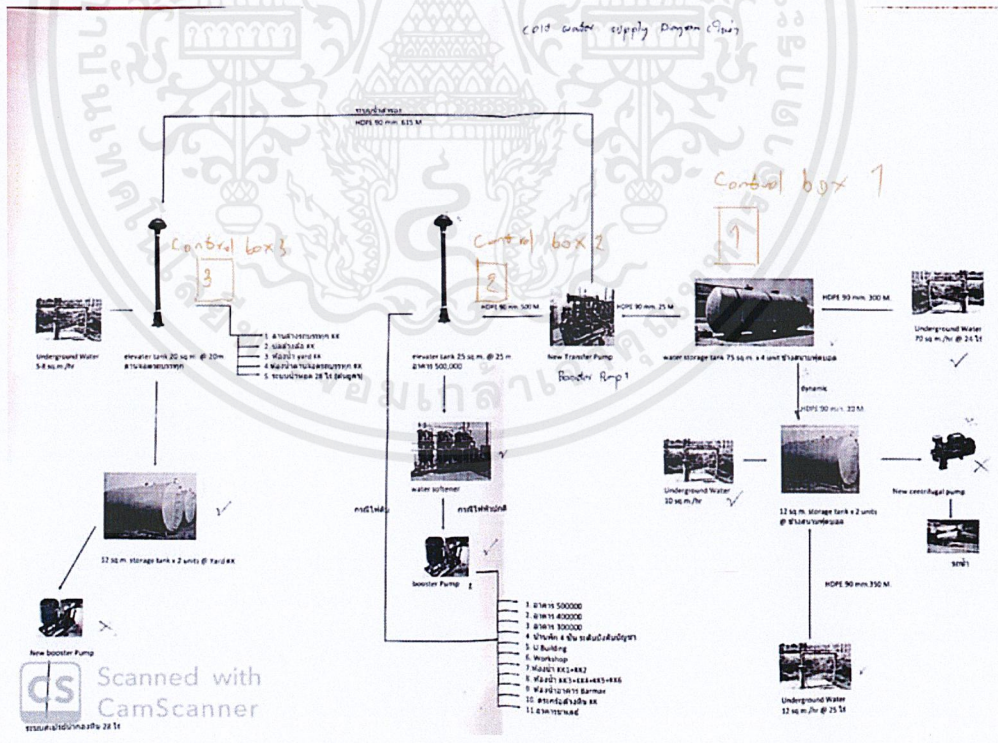
สวิตช์แรงดันใช้ในการควบคุมความดันในถังเก็บน้ำ โดยใช้การตัดหรือต่อวงจรไฟฟ้าเมื่อความดันถึงจุดที่เรากำหนดไว้



ภาพที่ 3.10 สวิตช์แรงดัน (Pressure Switch)

3.2.3 ออกแบบตู้ควบคุม

โดยจะติดตั้งตู้คอนโทรลบริเวณหน้างานจำนวน 3 ตู้ ตามภาพด้านล่าง




ภาพที่ 3.11 บริเวณที่ติดตั้งตู้คอนโทรล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 30
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3.1 เลือกขนาดของตู้คอนโทรล

โดยที่ควรเลือกขนาดตู้ให้เหมาะสมกับอุปกรณ์ภายใน โดยนักศึกษาได้เลือกเป็น ตู้กันฝุ่นขนาด 630×750×250 มิลลิเมตร จำนวน3ตู้ เพราะว่ามีขนาดที่พอเหมาะ และขนาดใหญ่พอที่ทำให้ระบบสายความ ร้อนได้ดี ทำให้สามารถยืดอายุการใช้งานของไมโครคอนโทรลเลอร์ได้



CBSG IP55

CBSG Standard enclosure has double door panels and a clear window on the front door panel for easy monitoring. This type of enclosure is suitable for outdoor purposes. With an IP55 rating, it can help to protect the electrical equipment inside the enclosure from dust and liquid. It is made of a cold rolled steel sheet and has 20 standard sizes.

Specification	
Material	Cold rolled steel sheet
Steel Thickness	• Size 1-20: Enclosure's body and mounting plate: 1.5 mm. Door panel: 1.5 mm.
Finished Surface	Polyester powder coated in RAL7032
Supplies	Mounting plate and bottom cable gland plate
Technical Data	• Folded enclosure's front lip for rigidity and a gutter to prevent liquid • Multi-folding prevents the ingress of dust and liquid when opened • Enclosure's canopy helps prevent water damage • Zinc alloy housing lock • Door opens up to 120 degrees • Removable cable gland plate at the bottom • TAMCO locking system
Protection Rating	IP55

ภาพที่ 3.12 Booster Pump & Control Box

ในการควบคุมระบบน้ำ ทางบริษัทต้องการให้ควบคุมได้ทั้งแบบ Auto และเมื่อมีเหตุฉุกเฉินก็สามารถควบคุมด้วยระบบ Manual ที่ตู้ควบคุมบริเวณหน้างานได้ โดยจะเปลี่ยน

- ระบบAUTO จะให้ Arduinoเป็นตัวสั่งการให้ หน้าสัมผัสของRelayเปลี่ยนสถานะ เมื่อได้รับสัญญาณ input
- ระบบMANUAL จะใช้การควบคุมผ่าน push button switch ในการสั่ง start หรือ stop ปุ่มน้ำ

3.2.3.2 ออกแบบหน้าตู้

ที่บริเวณหน้าตู้ของตู้คอนโทรล จะประกอบไปด้วยสัญญาณไฟเพื่อแจ้งเตือนสถานะต่างๆ Push botton switch Emergency switch และ selector switch โดยการออกแบบไฟสัญญาณจะออกแบบตามมาตรฐานด้านล่าง

1) ไฟสัญญาณ (Pilot Lamp)

เป็นไฟที่ใช้แสดงสถานะต่างๆที่บริเวณหน้าตู้คอนโทรลระบบน้ำจะมีดังนี้

การแจ้งเตือน	สี
READY	สีเขียว
FAULT	สีแดง
START	สีเขียว
STOP	สีเขียว
Level High	สีน้ำเงิน
Level Low	สีน้ำเงิน

ตารางที่ 3.2 ตารางบอกสีของไฟแจ้งเตือนที่ใช้

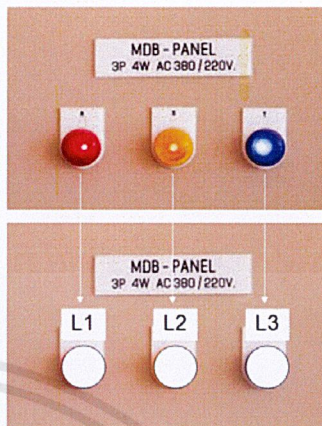


ภาพที่ 3.13 ไฟสัญญาณ(Pilot Lamp) สีต่างๆ

Color of Pilot lamp ?

Pilot Lights: IEC 60204-1

Color	Meaning	Explanation	Action by Operator
Red	Emergency	Hazardous condition	Immediate action to deal with hazardous condition (e.g., by operating the emergency-stop)
Yellow	Abnormal	Abnormal condition or impending critical condition	Monitoring and/or intervention (e.g., by reestablishing the intended function)
Green	Normal	Normal condition	Optional
Blue	Mandatory	Indication of a condition that requires action by the operator	Mandatory action
White	No specific meaning assigned	Other conditions; may be used whenever doubt exists about the application of other colors	Monitoring



ภาพที่ 3.14 มาตรฐานสีของไฟแจ้งเตือน

2) สวิตช์ปุ่มกด (Push Button Switch)

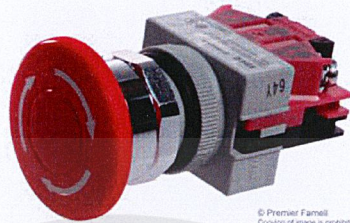
เป็นสวิตช์ปุ่มกดแบบ เมื่อกดแล้วจะเปลี่ยนสถานะแต่เมื่อปล่อยมือออกก็จะกลับสู่สถานะเดิม โดย Push Button Switch จะสามารถใช้งานได้ 2 โหมด MANUAL เท่านั้น หน้าสัมผัสที่ใช้จะ ใช้ทั้งแบบ ปกติเปิด(NO) ที่จะใช้ในการกด Start และ ปกติปิด(NC) ที่ใช้ในการ Stop



ภาพที่ 3.15 สวิตช์ปุ่มกด(Push Button Switch)

3) สวิตช์ดอกเห็ด (Emergency Switch)

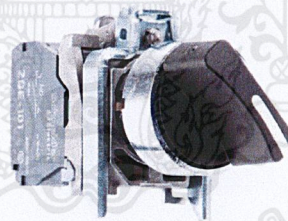
เป็นสวิตช์ปุ่มกดแบบจะค้างสถานะแม้จะปล่อยมือออกหน้าสัมผัสก็ยังคงสถานะไว้ โดยหน้าสัมผัสที่ใช้จะใช้เป็น ปกติ(NC) จะใช้งานเมื่อมีเหตุฉุกเฉินหรือเหตุขัดข้องจะทำการตัดไฟของคอนโทรลเลอร์



ภาพที่ 3.16 สวิตช์ดอกเห็ด(Emergency Stop)

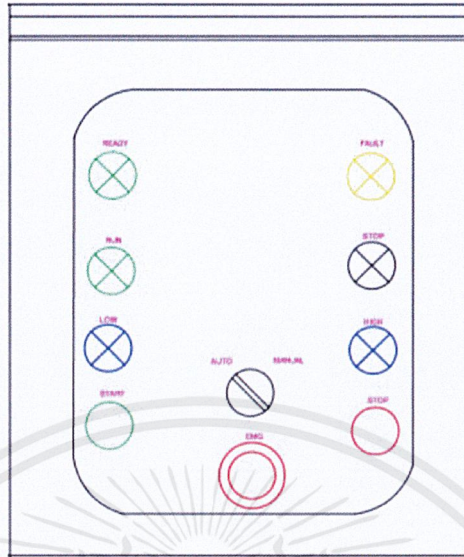
4) สวิตช์ตัวเลือก (Selector Switch)

Selector Switchที่ใช้จะใช้เป็นแบบ2ตำแหน่ง จะใช้หน้าสัมผัสแบบปกติเปิด(NO)ทำหน้าที่ในการเลือกโหมด AUTO หรือ MANUAL



ภาพที่ 3.17 สวิตช์ตัวเลือก (Selector Switch)

ภาพรวมของหน้าตู้คอนโทรล

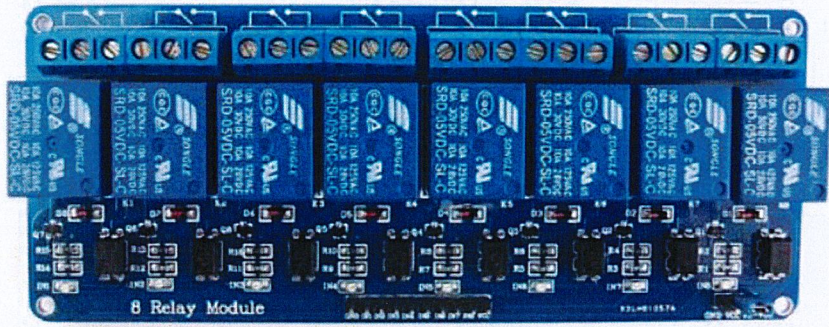


ภาพที่ 3.18 สวิตช์ตัวเลือก (Selector Switch)

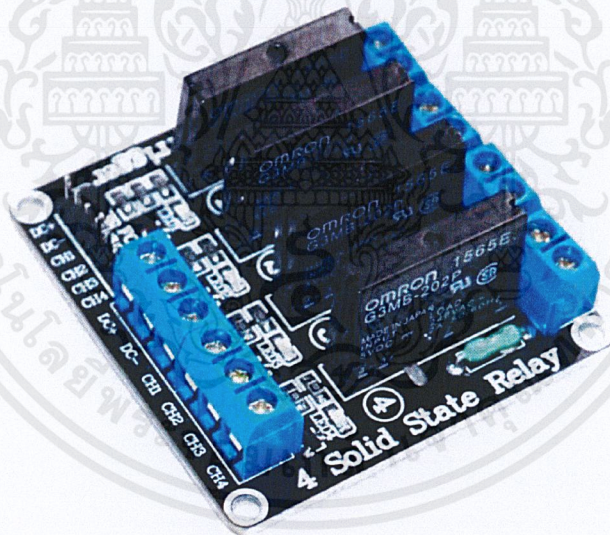
3.2.3.3 อุปกรณ์ที่ติดตั้งภายในตู้

1) รีเลย์(Relay)

รีเลย์ที่สามารถใช้ได้กับ Arduino Mega Wifi ATmega โดยอินพุทและเอาต์พุท ของคอนโทรลเลอร์จะรับและส่งสัญญาณเป็นไฟกระแสตรง 5 โวลต์ จึงได้เลือกใช้รีเลย์แบบ โซลิดสเตตมาเป็นรีเลย์เอาต์พุท เพราะว่าหน้าสัมผัสจะเปลี่ยนสถานะเมื่อมีสัญญาณไฟกระแสตรง 5 โวลต์ มากระตุ้นที่คอล์ย แต่รีเลย์ด้านอินพุทจะใช้เป็น โมดูลรีรีเลย์ 24 โวลต์ จะใช้ไฟกระแสตรง 24 โวลต์ ในการทำให้หน้าสัมผัสเปลี่ยนสถานะและในการส่งสัญญาณในระยะไกลหลายร้อยเมตรเมื่อส่งเป็นสัญญาณไฟแบบ 5 โวลต์ อาจทำให้เกิดค่าโวลต์ดรอปทำให้เกิดความผิดพลาดในการควบคุมได้



ภาพที่ 3.19 โมดูลรีเลย์ 24 โวลต์ 8 chanal



ภาพที่ 3.20 โมดูลรีเลย์โซลิตสเทท 5 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) เซอร์กิตเบรกเกอร์ (Circuit Breaker)

เป็นอุปกรณ์ป้องกันวงจร โดยจะเลือกใช้เป็นลูกเซอร์กิตเบรกเกอร์แบบ 2 ขั้ว และ 1 ขั้ว เพื่อป้องกันคอนโทรลเลอร์ไม่ให้เสียหายเมื่อเกิดการลัดวงจร



ภาพที่ 3.21 เซอร์กิตเบรกเกอร์ แบบ 2 ขั้ว (ซ้าย) ละ 1 ขั้ว (ขวา)

4) สวิตซ์ซิ่งเพาเวอร์ซัพพาย (Switching Powersupply)

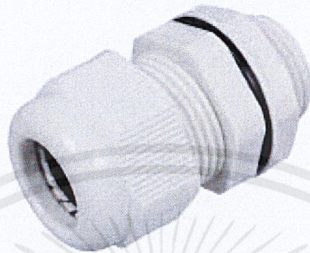
เป็นอุปกรณ์ในการจ่ายไฟกระแสตรงให้อุปกรณ์ต่างๆ โดยในตู้จะประกอบไปด้วยสวิตซ์ซิ่งเพาเวอร์ซัพพาย 2 ตัว มี 5 โวลต์ และ 24 โวลต์ โดยให้ตัว 5 โวลต์เป็นตัวจ่ายไฟเลี้ยง Arduino (5 Vdc) และเป็นตัวจ่ายแรงดันขาเข้าของ Arduino ส่วนตัว 24 โวลต์จะทำหน้าที่เป็นตัวจ่ายสัญญาณขาออก



ภาพที่ 3.22 สวิตซ์ซิ่งเพาเวอร์ซัพพาย (Switching Powersupply)

5) Cable Grand

ใช้เป็นทางเข้าออกของสายไฟของตู้คอนโทรล เพื่อป้องกันไม่ให้ฉนวนของสายไฟเสียหายและป้องกันสิ่งแปลกปลอมเข้าไปในภายในตู้คอนโทรล



ภาพที่ 3.23 Cable Grand

6) Cable Tie

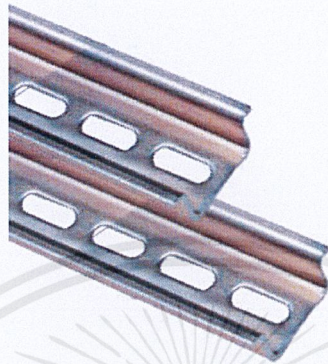
ใช้เพื่อจัดระเบียบของสายไฟไม่ให้ยุ่งเหยิง เพื่อความเป็นระเบียบและง่ายต่อการซ่อมบำรุง



ภาพที่ 3.24 Cable Tie

7) รางปีกนก (Omega rail)

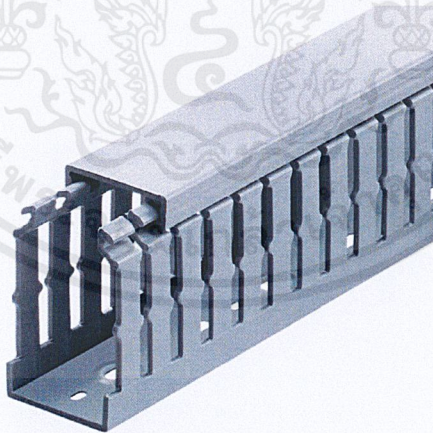
ใช้สำหรับติดตั้งอุปกรณ์ต่างๆภายในตู้คอนโทรล โดยจะใช้รางOmegaเป็นฐาน



ภาพที่ 3.25 รางปีกนก (Omega rail)

8) Cable Duct

เป็นอุปกรณ์ที่ใช้ซ่อนสายไฟเพื่อความเป็นระเบียบและสวยงาม



ภาพที่ 3.26 Cable Duct

9) สายคอนโทรล (Control Cable)

สายไฟขนาดเล็กใช้สำหรับ เชื่อมต่ออุปกรณ์ภายในตู้คอนโทรล

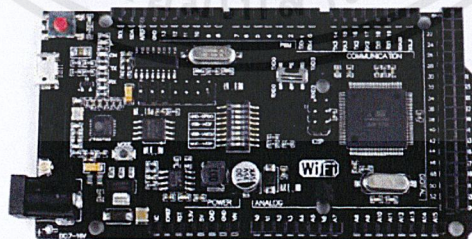


ภาพที่ 3.27 สายคอนโทรล (Control Cable)

3.2.4 เขียนโปรแกรม Arduino

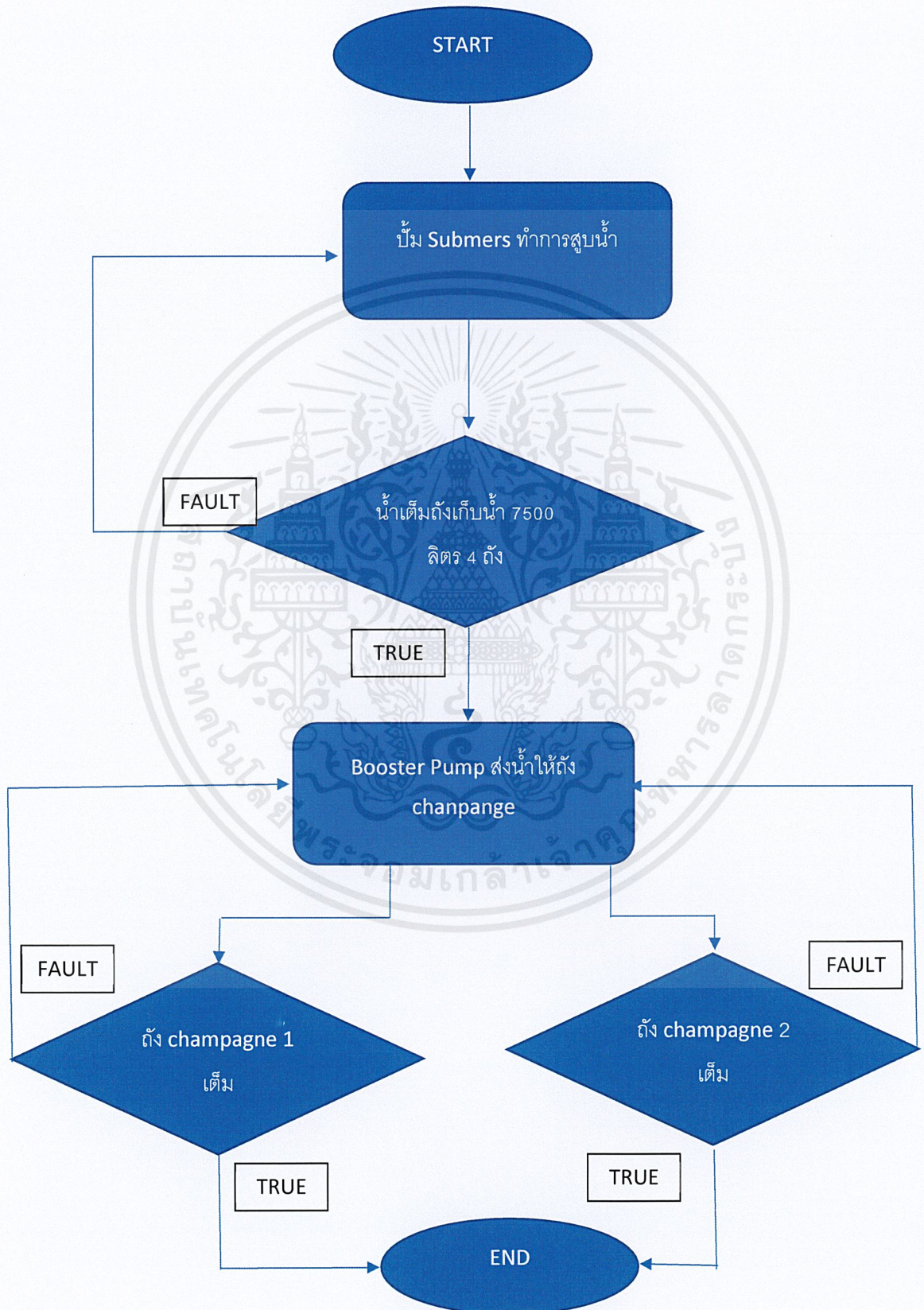
ในการเขียนโปรแกรมเพื่อนำโหนดให้กับบอร์ดอาร์ดูโน้ จะใช้ภาษาพื้นฐานจากC++ ซึ่งจะเขียนลงในโปรแกรม ARDUINO ซึ่งเป็น ซอฟต์แวร์จากผู้ผลิตผู้ผลิต

โดยArduinoที่ใช้จะเป็นรุ่น Arduino Mega Wifi ATmega2560+ESP8266



ภาพที่ 2.28 Arduino Mega Wifi ATmega2560+ESP8266

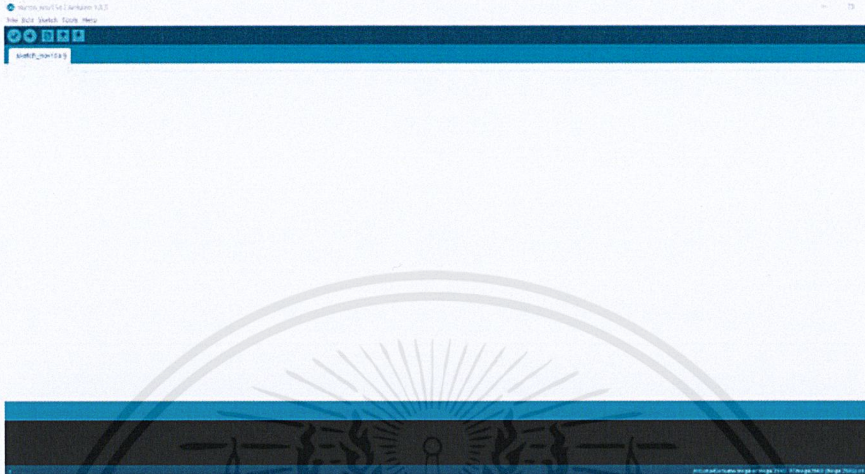
3.3.1 แผนผังการทำงาน (FLOW CHART)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 เขียนโปรแกรมลงใน ARDUINO

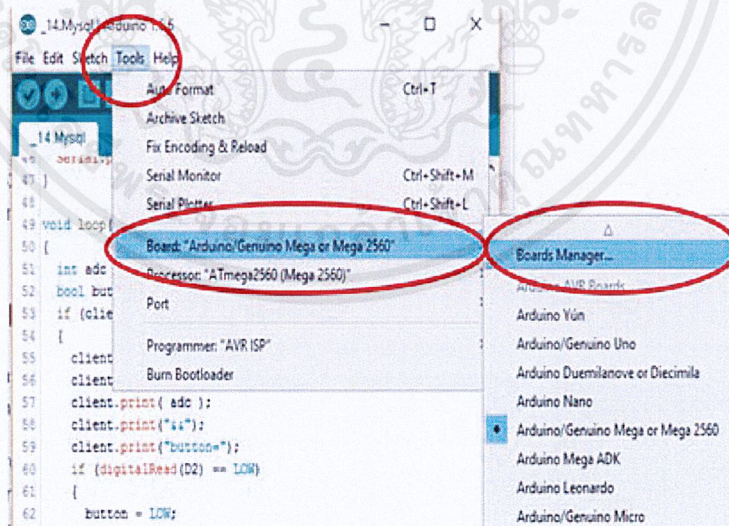
3.3.2.1) INTERFACE ของ Program Arduino IDE



ภาพที่ 3.29 Interface ของ Program Arduino IDE

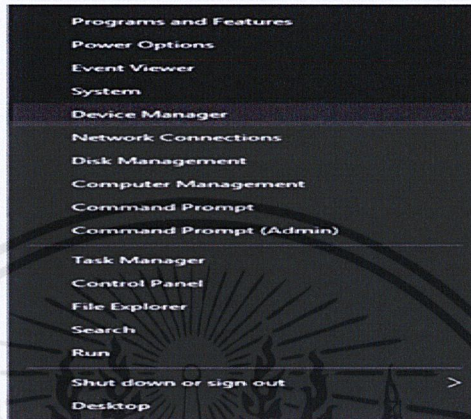
3.3.2.2) การเชื่อมต่อบอร์ด Arduino Arduino Mega Wifi กับคอมพิวเตอร์

1. ในหน้าต่างโปรแกรม Arduino ให้คลิกไปที่เมนู Tools -> Board -> Arduino Mega 2560



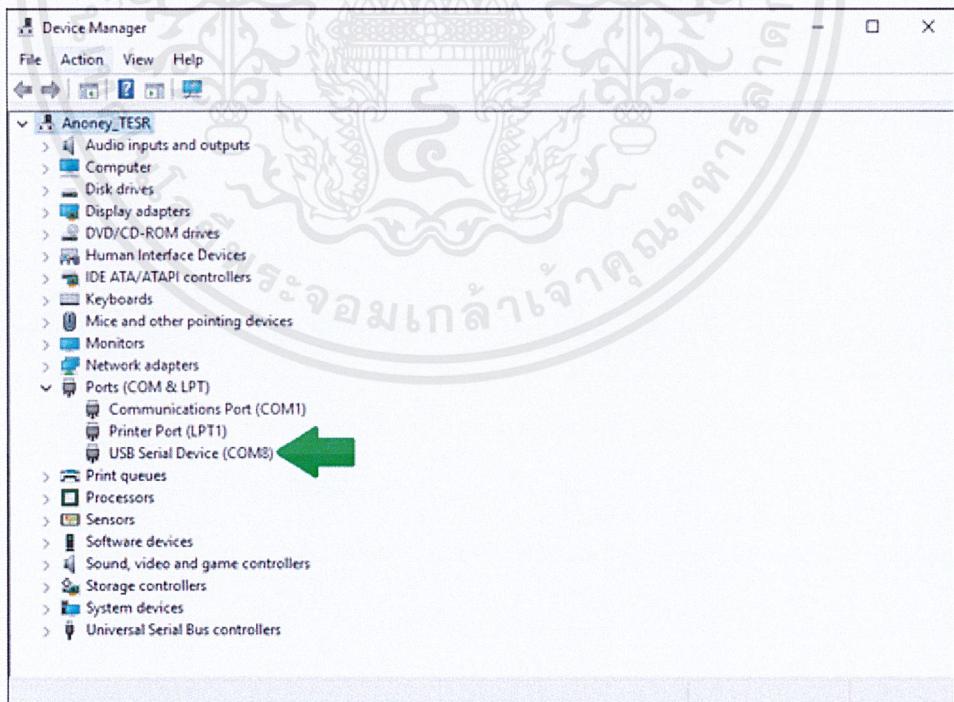
ภาพที่ 3.30 Interface ของ Program Arduino IDE

2.เสียบบอร์ด Arduino Mega 2560 เข้ากับคอมพิวเตอร์ จากนั้นไปที่ Device Manager (หากเป็น Window ให้คลิกขวาที่ไอคอนบน Toolbars) เพื่อดูว่าบอร์ด Arduino Mega 2560 นั้นต่ออยู่กับ COM Port หมายเลขใด



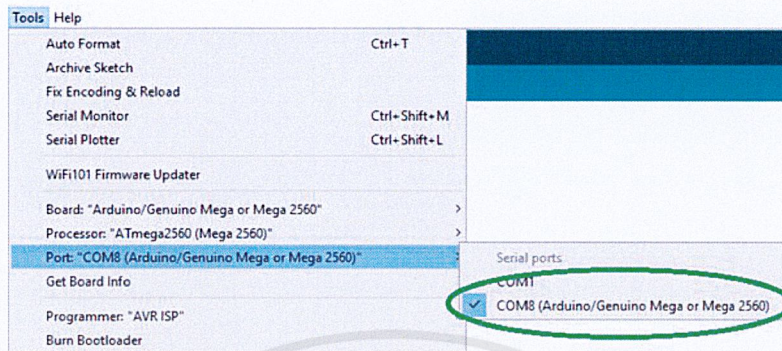
ภาพที่ 3.31 Interface ของ Program Arduino IDE

3.ใน Device Manager คลิกที่ Ports เพื่อดูหมายเลข COM Port ดังภาพ



ภาพที่ 3.32 Interface ของ Program Arduino IDE

4.การเปิด Arduino IDE ขึ้นมาอีกครั้ง ไปที่เมนู Tools และทำการตั้งค่าบอร์ดและหมายเลขพอร์ตให้ตรงกับที่ได้จากขั้นตอนที่ 3 ซึ่งในตัวอย่างคือ COM8 ดังนี้



ภาพที่ 3.33 Interface ของ Program Arduino IDE

3.3.2.3) เริ่มต้นเขียนโปรแกรม Arduino

1) ประกาศค่าตัวแปร เช่น ตัวแปร buttonpin1 ก็คือพอร์ทที่ 22 ของ Arduino

```
sketch_jan11a | Arduino 1.0.9
File Edit Sketch Tools Help
sketch_jan11a.g
int buttonPin1 = 22; //AUTO SUBMERGE
int buttonPin2 = 23; //OVERLOAD RELAY
int buttonPin3 = 24; //LEVEL LOW LOW 75Q
int buttonPin4 = 25; //LEVEL LOW 75Q
int buttonPin5 = 26; //LEVEL HIGH 75Q
int buttonPin6 = 27; //START MOTOR FROM TRUCK
int buttonPin7 = 28; //START MOTOR FROM TANK2
int buttonPin8 = 29; //START MOTOR FROM 500K
int buttonPin9 = 30; //MANUAL SUBMERGE
int buttonPin10 = 31; //START SUBMERGE
int buttonPin11 = 32; //STOP SUBMERGE
int ledPin1/READYLAMP = 52; //READY LAMP
int ledPin2/SUBRUNLAMP = 50; //SUBMERSE RUN LAMP
int ledPin3 = 40; //SUBMERGE STOP LAMP
int ledPin4 = 46; //FAULT LAMP
int ledPin11 = 53; //LEVEL LOW LOW 75Q LAMP
int ledPin5 = 44; //LEVEL LOW 75Q LAMP
int ledPin6 = 42; //LEVEL HIGH 75Q LAMP
int ledPin7 = 40; //BOOSTER PUMP RUN
int ledPin8 = 38; //BOOSTER PUMP STOP
int ledPin9 = 36; //AUTO***
int ledPin10 = 34; //MANUAL***
```

ภาพที่ 3.34 ภาพการกำหนดตัวแปรในโปรแกรม Arduino IDE

2) กำหนดให้แต่ละพอร์ทของArduino เป็น output หรือ input โดยใช้คำสั่ง pinMode(ตัวแปร ,output/input)

```
void setup() {  
  // ส่วนของการตั้งค่าประเภทอินพุตและเอาต์พุตของแต่ละขา  
  pinMode(ledPin1, OUTPUT);  
  pinMode(ledPin2, OUTPUT);  
  pinMode(ledPin3, OUTPUT);  
  pinMode(ledPin4, OUTPUT);  
  pinMode(ledPin5, OUTPUT);  
  pinMode(ledPin6, OUTPUT);  
  pinMode(ledPin7, OUTPUT);  
  pinMode(ledPin8, OUTPUT);  
  pinMode(ledPin9, OUTPUT);  
  pinMode(ledPin10, OUTPUT);  
  pinMode(ledPin11, OUTPUT);  
  //pinMode(buttonPin1, INPUT);  
  pinMode(buttonPin2, INPUT);  
  pinMode(buttonPin3, INPUT);  
  pinMode(buttonPin4, INPUT);  
  pinMode(buttonPin5, INPUT);  
  pinMode(buttonPin6, INPUT);  
  pinMode(buttonPin7, INPUT);  
  pinMode(buttonPin8, INPUT);  
  pinMode(buttonPin9, INPUT);  
  pinMode(buttonPin10, INPUT);  
  pinMode(buttonPin11, INPUT);  
  pinMode(buttonPin12, INPUT);  
}
```

ภาพที่ 3.35 การกำหนดOUTPUT INPUT ในโปรแกรม Arduino IDE

3) กำหนดให้OUTPUTเป็นHIGHเนื่องจากใช้ Relay Output เป็นแบบ Active Low เพื่อป้องกันไม่ให้ Relay ทำงาน

```
digitalWrite(ledPin1, HIGH);  
digitalWrite(ledPin2, HIGH);  
digitalWrite(ledPin3, HIGH);  
digitalWrite(ledPin4, HIGH);  
digitalWrite(ledPin5, HIGH);  
digitalWrite(ledPin6, HIGH);  
digitalWrite(ledPin7, HIGH);  
digitalWrite(ledPin8, HIGH);  
digitalWrite(ledPin9, HIGH);  
digitalWrite(ledPin10, HIGH);  
digitalWrite(ledPin11, HIGH);
```

ภาพที่ 3.36 กำหนดให้ OUTPUT มีสถานะเป็น HIGH

4) ส่วนของการทำงานแบบ Loop โปรแกรมจะทำการตรวจสอบเงื่อนไขของโปรแกรมนวนไปเรื่อยๆ

```
void loop() {
//buttonStatus1 = digitalRead(buttonPin1); //FLOW SENSOR
buttonStatus2 = digitalRead(buttonPin2); //OVERLOAD RELAY
buttonStatus3 = digitalRead(buttonPin3); //LEVEL LOW
buttonStatus4 = digitalRead(buttonPin4); //LEVEL HIGH
buttonStatus5 = digitalRead(buttonPin5); //START MOTOR FROM 500K
buttonStatus6 = digitalRead(buttonPin6); //START MOTOR FROM TRUCK
buttonStatus7 = digitalRead(buttonPin7); //START MOTOR FROM TANK2
buttonStatus8 = digitalRead(buttonPin8); //AUTO SUBMERGE
buttonStatus9 = digitalRead(buttonPin9); //MANUAL SUBMERGE
buttonStatus10 = digitalRead(buttonPin10); //START SUBMERGE
buttonStatus11 = digitalRead(buttonPin11); //STOP SUBMERGE
//buttonStatus12 = digitalRead(buttonPin12); //LEVEL LOW LOW 75Q

//*****AUTO*****
if (buttonStatus8 == HIGH && buttonStatus9 == LOW) //AUTO ON,MANUAL OFF
{
digitalWrite(ledPin9, LOW); //AUTO LAMP ON
digitalWrite(ledPin10, HIGH); //MANUAL LAMP OFF
buttonStatus3 = digitalRead(buttonPin3); //LEVEL LOW
buttonStatus4 = digitalRead(buttonPin4); //LEVEL HIGH
// buttonStatus1 = digitalRead(buttonPin1); //FLOW SENSOR
buttonStatus2 = digitalRead(buttonPin2); //OVERLOAD RELAY
buttonStatus5 = digitalRead(buttonPin5); //START MOTOR FROM 500K
buttonStatus6 = digitalRead(buttonPin6); //START MOTOR FROM TRUCK
buttonStatus7 = digitalRead(buttonPin7); //START MOTOR FROM TANK2
// buttonStatus12 = digitalRead(buttonPin12); //LEVEL LOW LOW 75Q
}
```

ภาพที่ 3.37 การทำงานแบบวนซ้ำ (LOOP)


5) คอมไพล์ (Compile) โดยคลิกที่ปุ่ม  เพื่อตรวจสอบว่าโค้ดที่เขียนไม่มีข้อผิดพลาด หากไม่มีข้อผิดพลาด จะปรากฏข้อความว่า “Done compiling” ดังภาพ

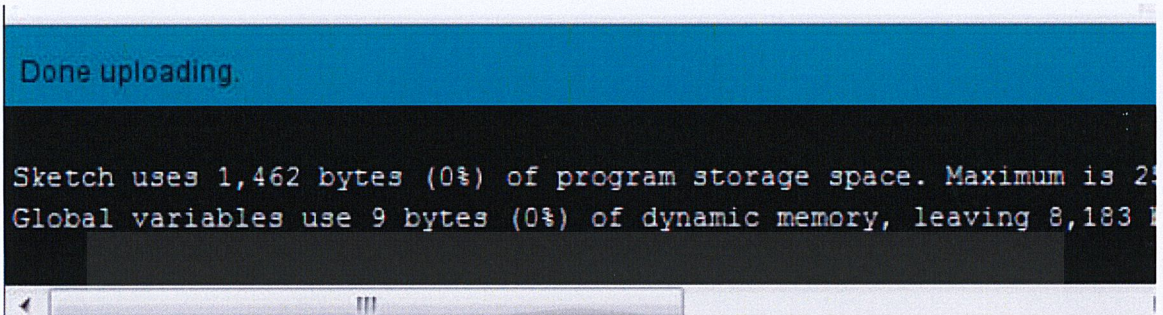
```
buttonStatus10 = digitalRead(buttonPin10); //START SUBMI
buttonStatus11 = digitalRead(buttonPin11); //STOP SUBMEI

Done compiling.

Sketch uses 2514 bytes (0%) of program storage space. Ma
Global variables use 29 bytes (0%) of dynamic memory, le
65
```

ภาพที่ 3.38 การตรวจสอบความถูกต้องของไวยากรณ์ (COMPILE)

6) จากนั้นให้ทำการคลิกที่ปุ่ม  เพื่อทำการอัปโหลดโค้ดเข้าสู่บอร์ด Arduino หากไม่มีข้อผิดพลาด จะขึ้นคำว่า “Done uploading” ดังภาพ



ภาพที่ 3.39 การอัปโหลดโปรแกรมลงไปในบอร์ดArduino

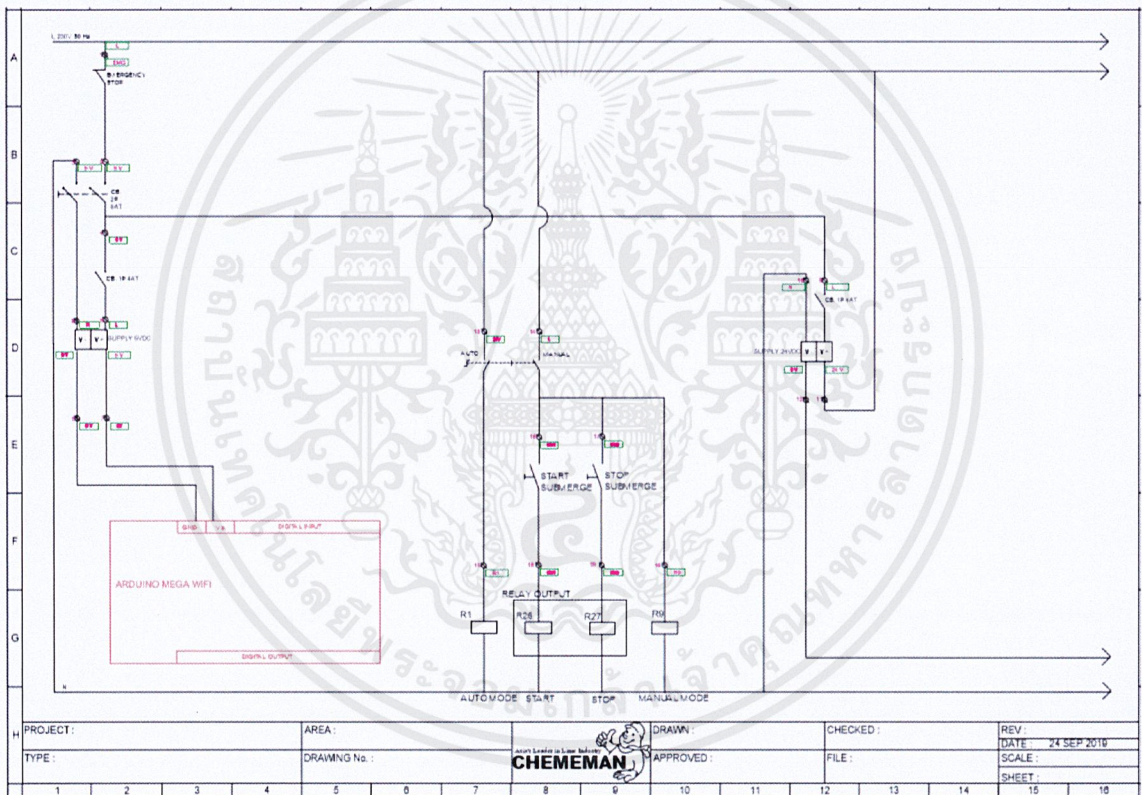


บทที่ 4

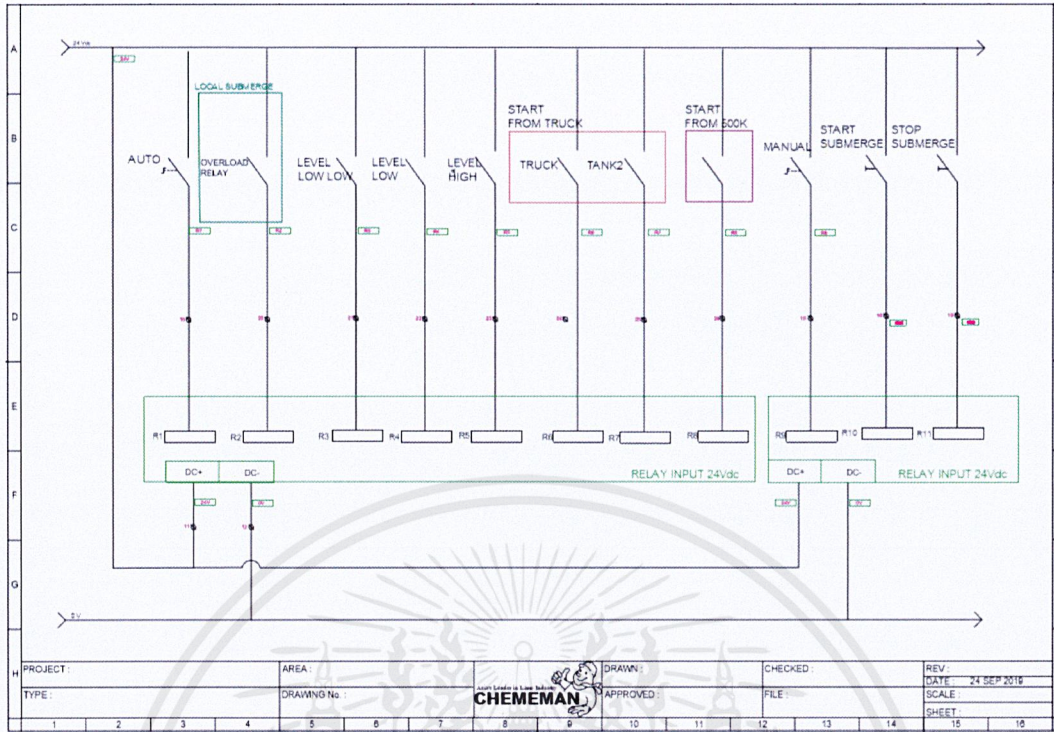
ผลการดำเนินงาน

4.1 แบบวงจรควบคุมระบบน้ำ

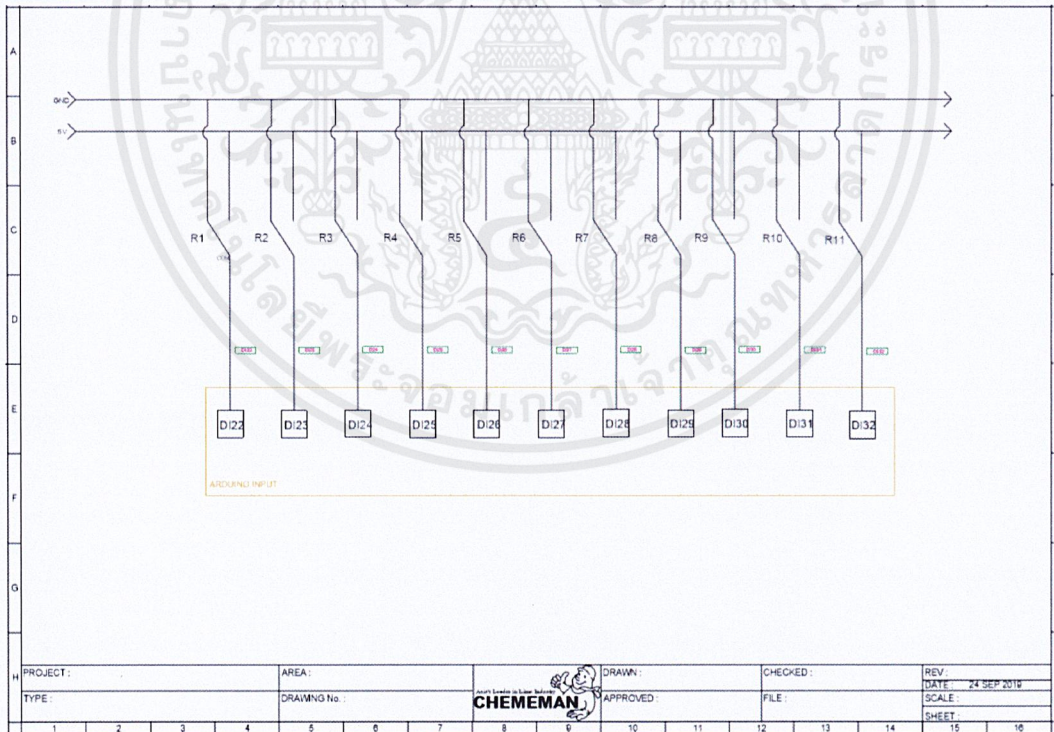
4.1.1 แบบวงจรควบคุมระบบของตู้ควบที่1 (บริเวณถังเก็บน้ำ 7500 ลิตร)



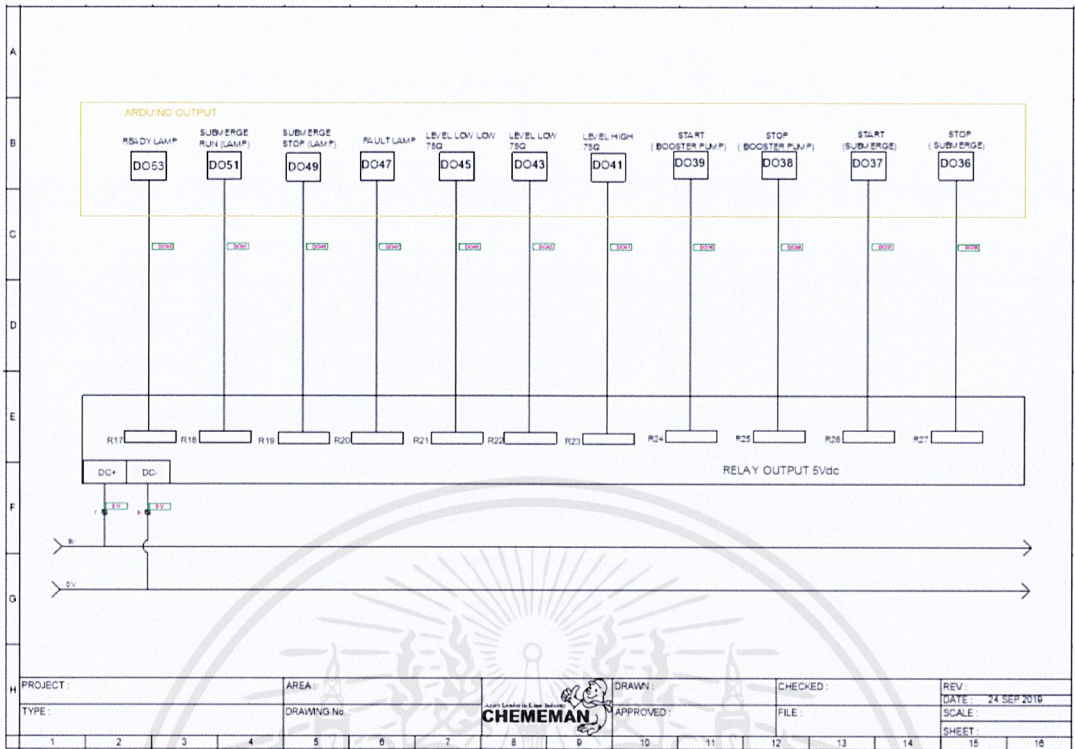
ภาพที่ 4.1 แบบวงจรควบคุมตู้ที่1 (1)



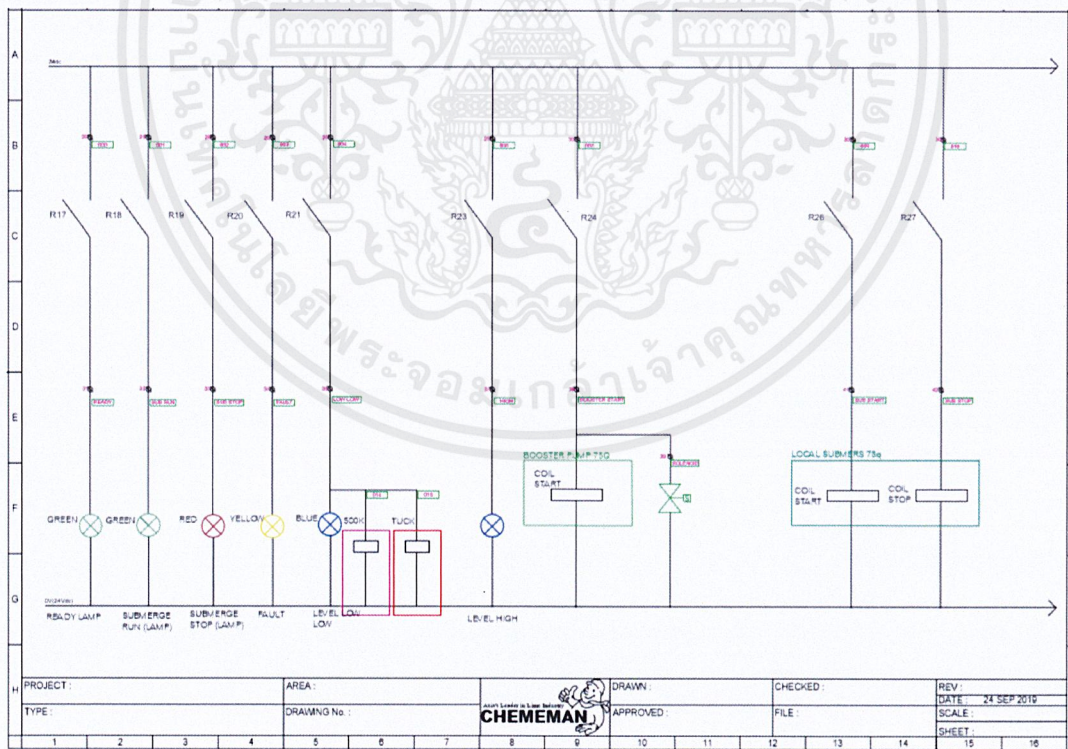
ภาพที่ 4.2 แบบวงจรควบคุมตู้ที่ 1 (2)



ภาพที่ 4.3 แบบวงจรควบคุมตู้ที่ 1 (3)

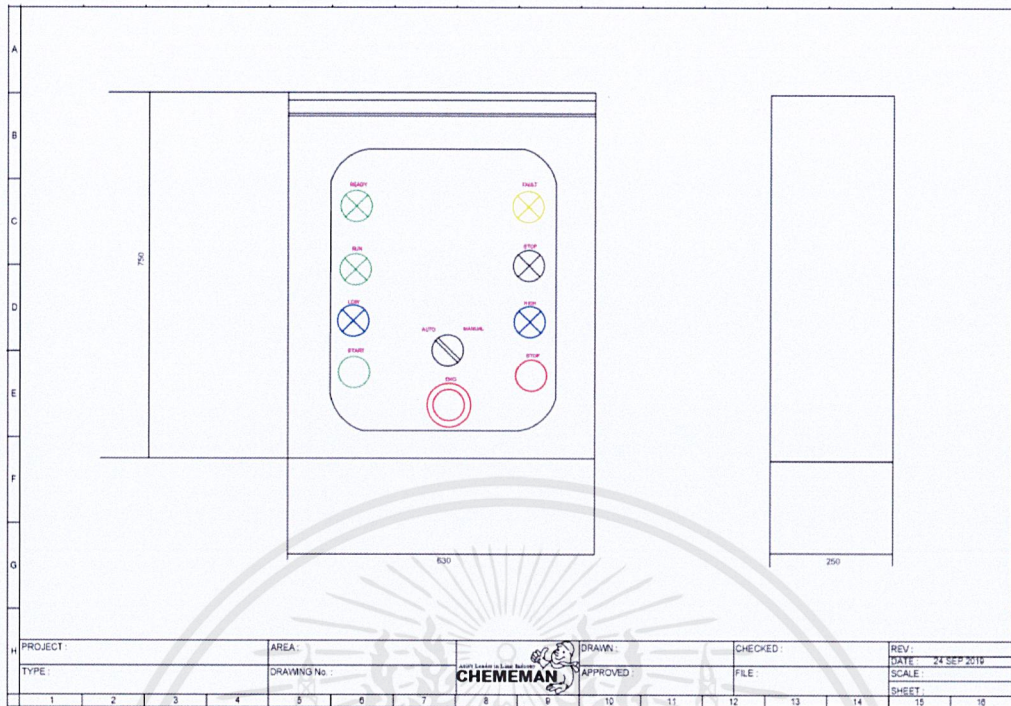


ภาพที่ 4.4 แบบวงจรควบคุมตู้ที่ 1 (4)



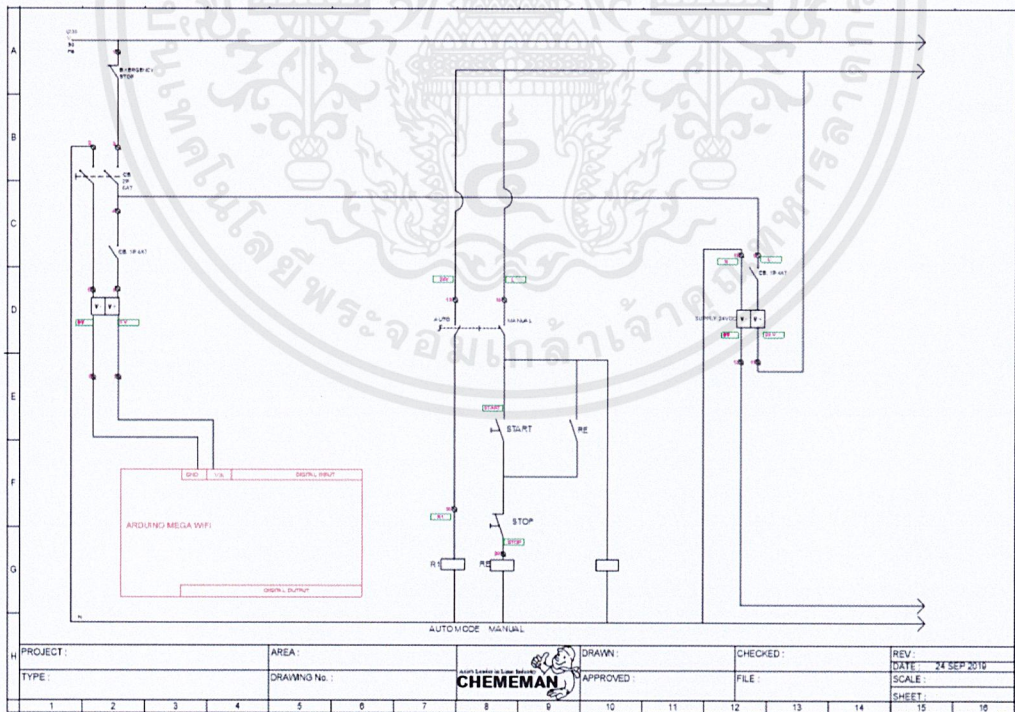
ภาพที่ 4.5 แบบวงจรควบคุมตู้ที่ 1 (5)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 50
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

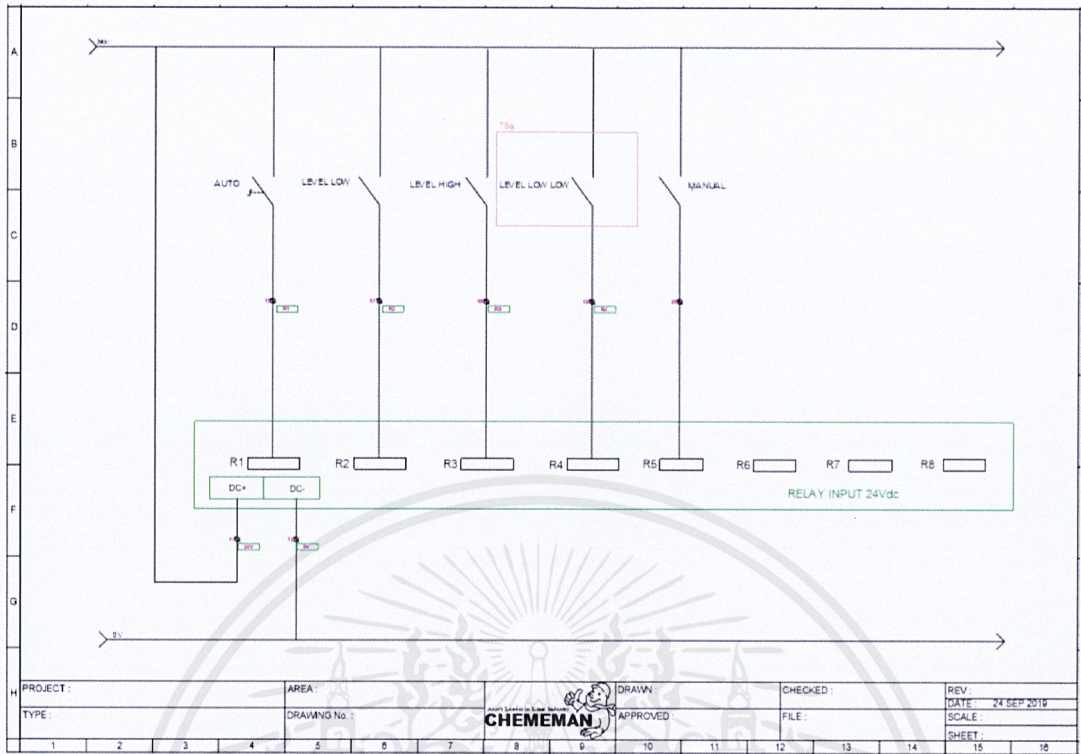


ภาพที่ 4.6 แบบวงจรควบคุมตู้ที่ 1 (6)

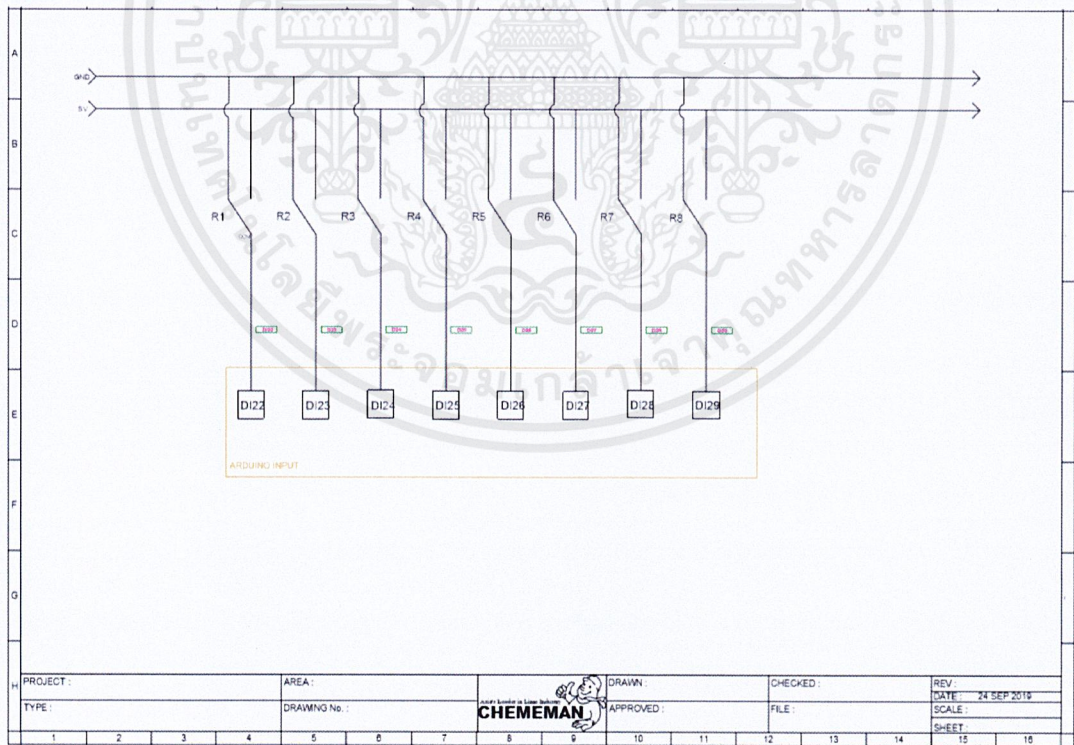
4.1.2 แบบวงจรควบคุมระบบของตู้ควบคุมที่ 2 (บริเวณอาคาร 500,000)



ภาพที่ 4.7 แบบวงจรควบคุมตู้ที่ 2 (1)

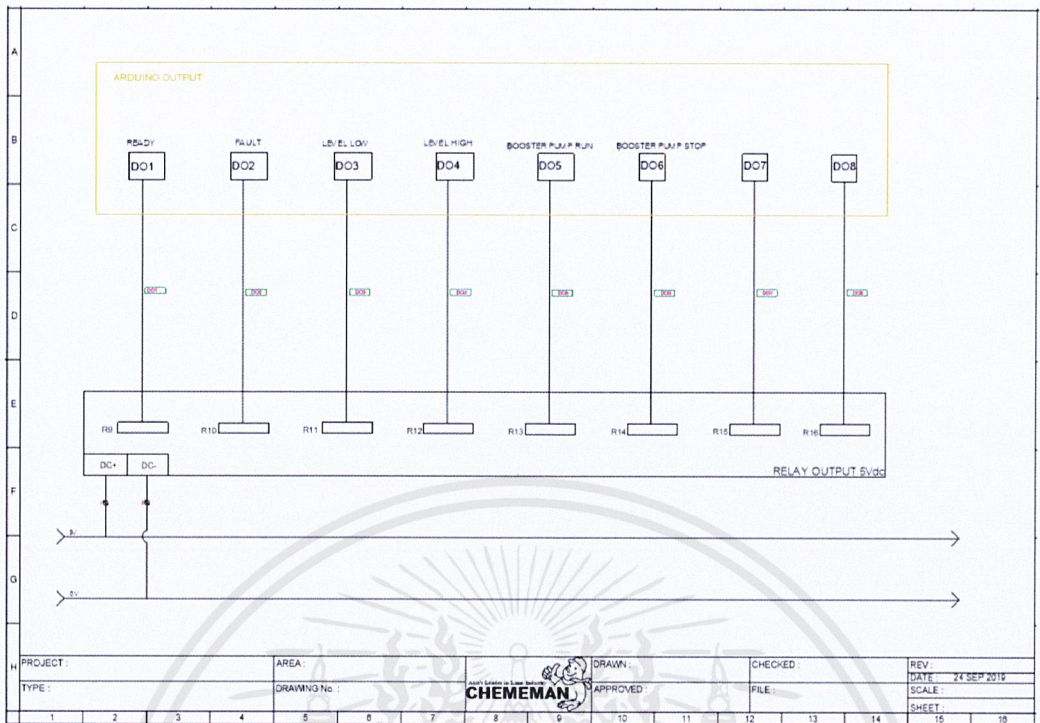


ภาพที่ 4.8 แบบวงจรควบคุมตู้ที่ 2 (2)

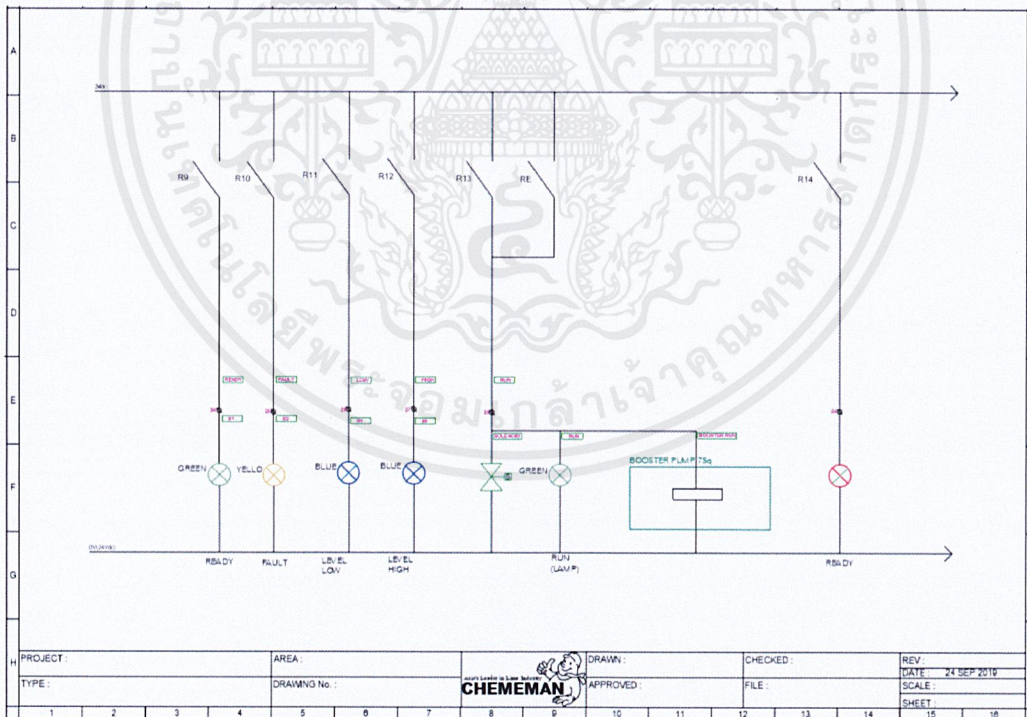


ภาพที่ 4.9 แบบวงจรควบคุมตู้ที่ 2 (3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 52
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

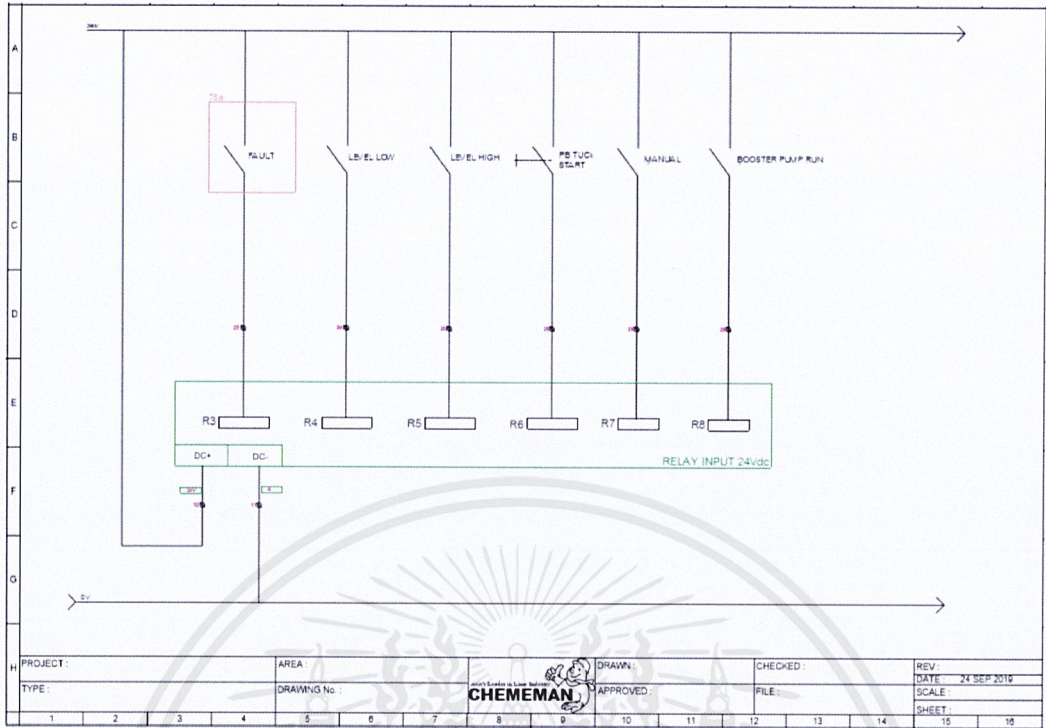


ภาพที่ 4.10 แบบวงจรควบคุมตู้ที่ 2 (4)

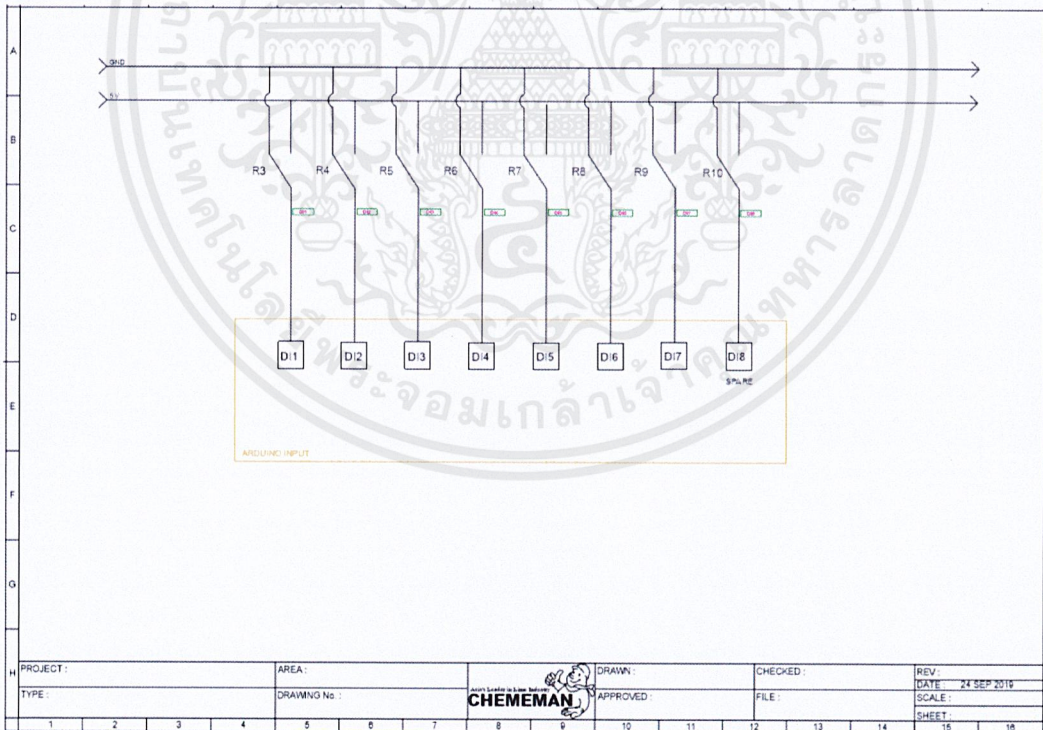


ภาพที่ 4.11 แบบวงจรควบคุมตู้ที่ 2 (5)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 53
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

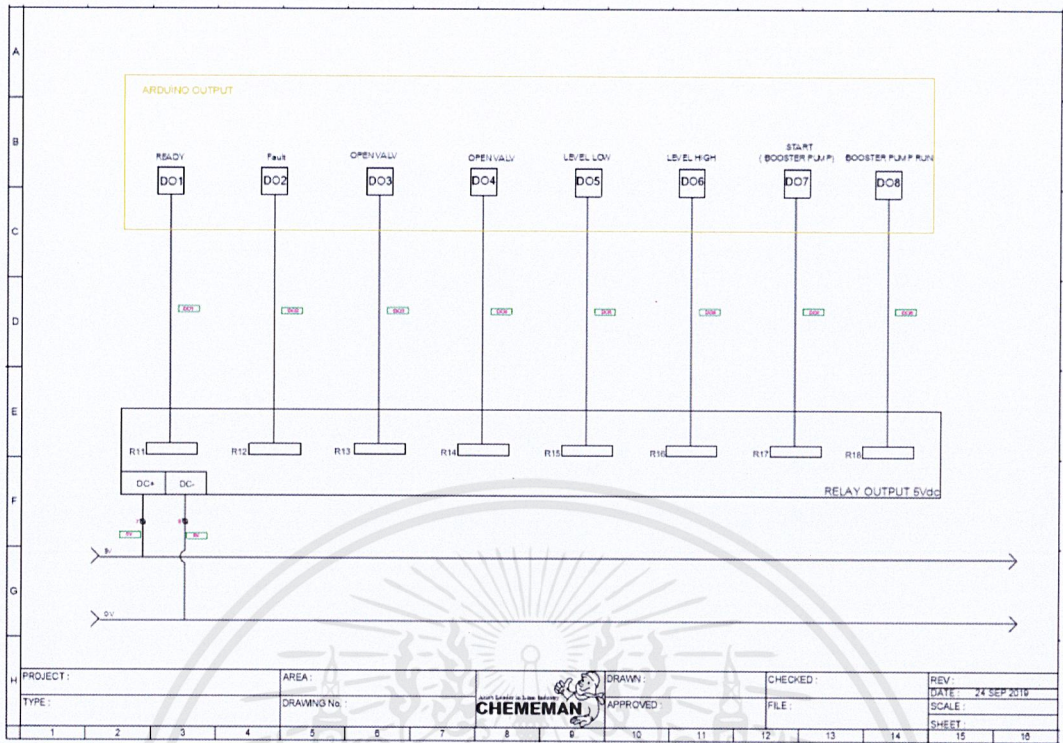


ภาพที่ 4.14 แบบวงจรควบคุมตู้ที่ 3 (2)

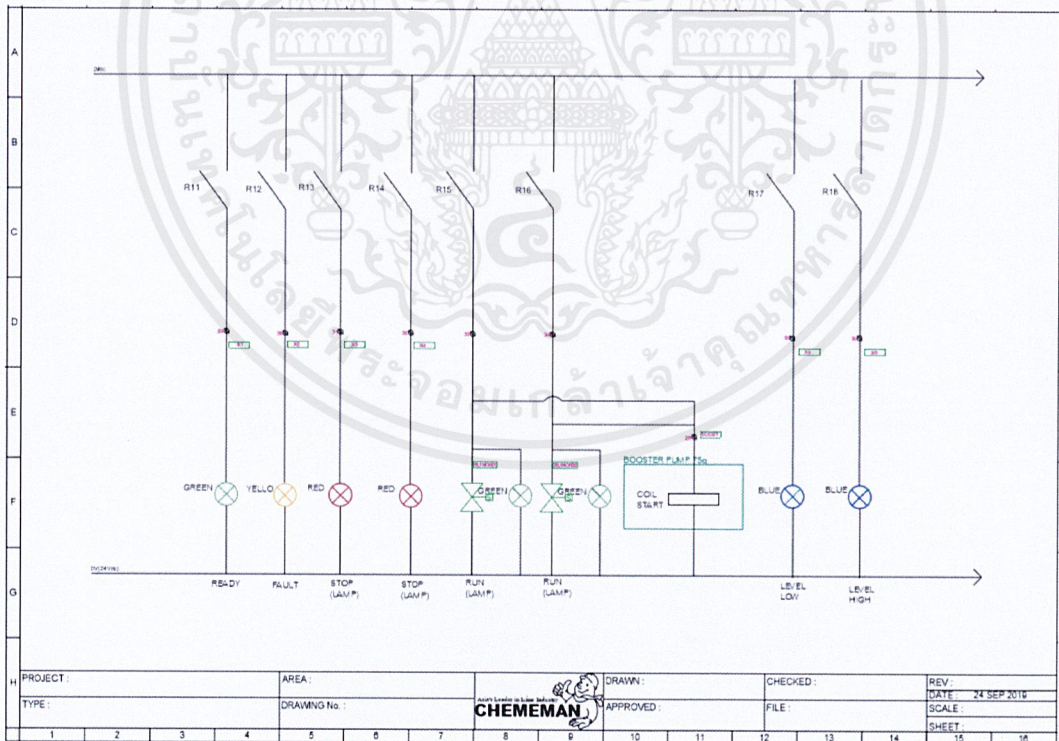


ภาพที่ 4.15 แบบวงจรควบคุมตู้ที่ 3 (3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 55
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

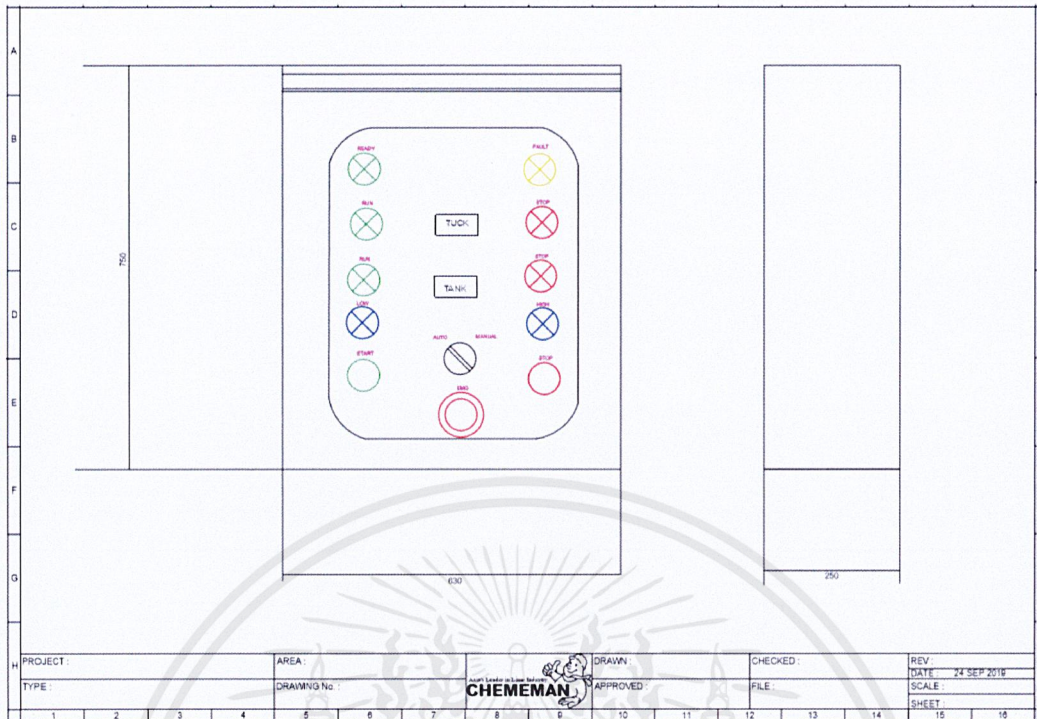


ภาพที่ 4.16 แบบวงจรควบคุมตู้ที่ 3 (4)



ภาพที่ 4.17 แบบวงจรควบคุมตู้ที่ 3 (5)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 56
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

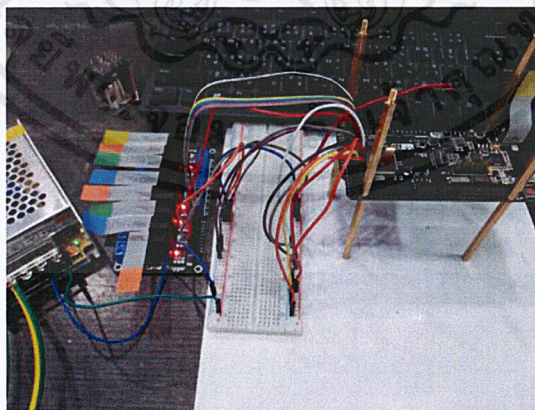


ภาพที่ 4.18 แบบวงจรควบคุมตู้ที่ 3 (6)

4.2 โปรแกรมควบคุม Arduino

4.2.1 ทำงานทดสอบโปรแกรมที่อัฟโพลดลงบอร์ด Arduino

เมื่อทำการอัฟโพลดโปรแกรมที่สร้างไว้ลงในบอร์ด Arduino แล้วก็ได้นำมาทดสอบโดยการจ่ายไฟ 5 โวลต์เพื่อนเป็นสัญญาณขาเข้า(input) แล้วดูการตอบสนองของ รีเลย์ว่าทำงานตามเงื่อนไขหรือไม่



ภาพที่ 4.19 การทดสอบการทำงานของ Arduino

INPUT		OUTPUT *(H=HIGH ,L=LOW)							
		ledPin1(52) READYLAMP	ledPin2(50) SUBRUNLAMP	ledPin3(48) SUBSTOPLAMP	ledPin4(46) FAULTLAMP	ledPin6(44) Level L	ledPin7(42) Level H	ledPin8(40) boost run	ledPin9(38) boost stop
buttonPin1(22) Auto		L	H	L	H	H	H	H	L
buttonPin2(23) Over Load		H	H	L	L	H	H	H	L
buttonPin3(24) Level LL		L	L	H	H	L	H	H	L
buttonPin4(25) Level L		L	L	H	H	L	H	H	L
buttonPin5(26) Level H		L	H	L	H	H	L	H	L
buttonPin6(27) Start Truck		L	H	L	H	L	H	L	H
buttonPin7(28) Start Tank2		L	H	L	H	L	H	L	H
buttonPin8(29) Start Tank1		L	H	L	H	L	H	L	H
buttonPin9(30) Manual		L	H	L	H	L	H	H	L
buttonPin10(31) Start SUB		L	L	H	H	H	H	H	L
buttonPin11(32) Stop SUB		L	H	L	H	H	H	H	L

ตารางที่ 4.1 การทดสอบการทำงานของArduino NO.1

จากการทดสอบ :ได้ผลลัพธ์ตรงตามเงื่อนไขที่กำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		OUTPUT *(H=HIGH ,L=LOW)				
		ledPin(52) RUN (AUTO MODE)	ledPin2(50) READY LAMP	ledPin3(48) FAULTLAMP	ledPin4(46) Level LAMP	ledPin5(44) Level High LAMP
INPUT	buttonPin1(27) AUTO	H	L	H	H	H
	buttonPin2(28) MANUAL	H	L	H	H	H
	buttonPin3(22) BUTTON START	L	L	H	H	H
	buttonPin4(23) BUTTON STOP	H	L	H	H	H
	buttonPin5(24) LEVEL LOW	L	L	H	L	H
	buttonPin6(25) LEVEL HIGH	H	L	H	H	L
	buttonPin7(26) FAULT form75Q	H	H	L	H	H

ตารางที่ 4.2 การทดสอบการทำงานของARDUINO NO.2

จากการทดสอบ :ได้ผลลัพธ์ตรงตามเงื่อนไขที่กำหนดไว้

OUTPUT *(H=HIGH ,L=LOW)							
	ledPin52) RUN	ledPin4(50) Ready	ledPin5(48) Fault	ledPin7(44) Level H	ledPin7(44) Level L	ledPin8(38) Truck Start	ledPin9(36) Truck Stop
buttonPin1(27) AUTO	H	L	H	H	L	H	L
buttonPin2(28) MANUAL	H	L	H	H	L	H	L
buttonPin3(22) Tank Start	L	L	H	H	L	H	L
buttonPin4(23) Tank Stop	H	L	H	H	L	H	L
buttonPin5(24) Level Low	L	L	H	L	H	H	L
buttonPin7(25) FAULT	H	H	L	H	L	H	L
buttonPin8(29) Truck Start	L	L	H	H	L	L	H
buttonPin9(30) Truck Stop	H	L	H	H	L	H	L

ตารางที่ 4.3 การทดสอบการทำงานของARDUINO NO.3

จากการทดสอบ :ได้ผลลัพธ์ตรงตามเงื่อนไขที่กำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุป ปัญหา ข้อเสนอแนะ

5.1 สรุป

จากการที่ได้ปฏิบัติงานตามที่ได้รับมอบหมายในแต่ละวันทำให้ได้เรียนรู้ประสบการณ์ในการทำงานระบบไฟฟ้า,งานซ่อมบำรุง,งานตรวจเช็คประจำปี และ งานสอบเทียบเครื่องมือวัด

จากการที่ได้ศึกษา และปฏิบัติโครงการระบบน้ำระบบใหม่ นักศึกษาได้ออกแบบวงจรควบคุมสำหรับโครงการระบบน้ำ โดยใช้ Arduino ควบคุมการทำงานของ มอเตอร์ โซลินอยวาล์ว และ อื่น ๆ จากนั้นทำการเขียนโปรแกรมเพื่อควบคุม Arduino จากนั้นการทดสอบผลปรากฏว่าได้ผลลัพธ์เป็นไปตามเงื่อนไขที่กำหนดไว้

5.2 ข้อเสนอแนะ

- 1) เนื่องจากการสั่งซื้ออุปกรณ์ในการทำโครงการของบริษัทนั้นต้องผ่านหลายขั้นตอนทำให้การสั่งซื้ออุปกรณ์ ล้าช้า
- 2) ตัวแทนจำหน่ายอุปกรณ์ไฟฟ้า จัดส่งสินค้าให้ล่าช้ากว่ากำหนดทำให้นักศึกษาทำโครงการไม่เสร็จตามที่กำหนดไว้
- 3) เนื่องจากนักศึกษาปฏิบัติงานอยู่ที่แผนกซ่อมบำรุงไฟฟ้าและวัดคุม ทำให้มีงานประจำวันมากส่งผลให้มีเวลาในการทำโครงงานน้อย
- 4) ในการออกแบบระบบไฟฟ้าและระบบควบคุม นักศึกษาต้องศึกษาและทำความเข้าใจภาษาที่ใช้ในการเขียนโปรแกรม รวมถึงหลักการออกแบบและการเลือกใช้อุปกรณ์ไฟฟ้าต่างๆ เพื่อให้สามารถดำเนินการได้อย่างมีประสิทธิภาพ

บรรณานุกรม

- [1] krusomkuan. 2556. “การควบคุมอัตโนมัติทางอุตสาหกรรม”. [ระบบออนไลน์]. แหล่งที่มา : <https://krusomkuan.weebly.com/364836093639365736293627363436103607364836193637361836091.html>
- [2] PSTGroup. 2560. “มอเตอร์ไฟฟ้า”. [ระบบออนไลน์]. แหล่งที่มา : <http://www.pstgroup.biz/articles/752/>
- [3] นายวรพงษ์ วงศ์มะณี. 2560. “มอเตอร์เหนี่ยวนำสามเฟสแบบกรงกระรอก”. [ระบบออนไลน์]. แหล่งที่มา : <https://saiminstument.wordpress.com/นายวรพงษ์-วงศ์มะณี-เรือ/>
- [4] Factomart. 2555. “วิธีการเริ่มเดินมอเตอร์”. [ระบบออนไลน์]. แหล่งที่มา : <https://mall.factomart.com/guide-to-motor-control/why-different-starting-methods/>
- [5] voltimum. 2558. “Soft Start ABB รุ่น PSTX”. [ระบบออนไลน์]. แหล่งที่มา : <https://www.voltimum.com.au/articles/abb-pstx-softstarters>
- [6] Indiamart. 2557. “Inverter ABB รุ่น ACS550”. [ระบบออนไลน์]. แหล่งที่มา : <https://www.indiamart.com/proddetail/abb-white-ac-drive-19645805988.html>
- [7] ว่าที่ ร.ต. อาจศึก มามีกุล. (2554). “โครงการฝึกอบรมเทคนิคการตรวจซ่อมและบำรุงรักษาระบบปั้มน้ำ ประปาเบื้องต้น”. สกลนคร : มหาวิทยาลัยราชภัฏสกลนคร
- [8] Factomart. 2558. “อุปกรณ์ไฟฟ้า”. [ระบบออนไลน์]. แหล่งที่มา : <https://www.factomart.com>
- [9] MCPBlogs ,2560 “ARDUINO”. [ระบบออนไลน์]. แหล่งที่มา : <http://blog.mcp.ac.th/?p=166>
- [10] iochop , 2559 “ภาษาที่ใช้เขียน Arduino”. [ระบบออนไลน์]. แหล่งที่มา <https://www.ioxhop.com/article/5/arduino>

ประวัติผู้เขียน

ชื่อ – นามสกุล : นาย อภิสิตธิ์ เต็มคุณานนท์

วัน เดือน ปีเกิด : 10 ธันวาคม 2540

ภูมิลำเนา : 34 หมู่ 11 ต.ลำเหย อ.ดอนตูม จ.นครปฐม 73150

E-mail : Apisit.termkunanon@gmail.com

โทรศัพท์ : 091 783 9517

ประวัติการศึกษา : ระดับมัธยม โรงเรียนพระปฐมวิทยาลัย

ระดับปริญญาตรี : ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประสบการณ์ : นักศึกษาฝึกงาน แผนก วิเคราะห์และวางแผน การไฟฟ้าส่วนภูมิภาค เขต ภาคกลาง 3

: นักศึกษาโครงการสหกิจศึกษา แผนกซ่อมบำรุงไฟฟ้าและวัดคุม บริษัท เคมีแมน จำกัด

(มหาชน)