

การปรับปรุงระบบการจุ่มเคลือบควบคุมด้วย  
ไมโครคอนโทรลเลอร์  
IMPROVEMENT OF DIP-COATING SYSTEM  
BASED ON MICROCONTROLLER



โครงงานพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร  
ปริญญาวิทยาศาสตรบัณฑิต (ฟิสิกส์ประยุกต์)  
ภาควิชาฟิสิกส์ คณะวิทยาศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2559

การปรับปรุงระบบการจุ่มเคลือบควบคุมด้วย  
ไมโครคอนโทรลเลอร์

IMPROVEMENT OF DIP-COATING SYSTEM  
BASED ON MICROCONTROLLER



T149469



b. 1288473 X  
i. ....

สงวนลิขสิทธิ์  
เลขทะเบียน 149469  
วันเดือนปี - 8 มี.ค. 2561

โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร  
ปริญญาวิทยาศาสตรบัณฑิต (ฟิสิกส์ประยุกต์)  
ภาควิชาฟิสิกส์ คณะวิทยาศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ **ปีการศึกษา 2559** ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IMPROVEMENT OF DIP-COATING SYSTEM  
BASED ON MICROCONTROLLER



A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF  
THE REQUIREMENT FOR  
THE DEGREE OF BACHELOR OF SCIENCE (APPLIED PHYSICS)  
DEPARTMENT OF PHYSICS FACULTY OF SCIENCE  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาและค้นคว้าเท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงการพิเศษ      การปรับปรุงระบบการจุ่มเคลือบควบคุมด้วยไมโครคอนโทรลเลอร์  
 Improvement of Dip-Coating System Based on Microcontroller

ชื่อนักศึกษา                    นางสาวศิริพร ทองเชื้อ      รหัสนักศึกษา 56051220  
    นางสาวอุไรพร วิหกหงษ์      รหัสนักศึกษา 56051254

ปริญญา                            วิทยาศาสตร์บัณฑิต (ฟิสิกส์ประยุกต์)  
 ภาควิชา                            ฟิสิกส์

ปีการศึกษา                      2559

อาจารย์ที่ปรึกษา                รศ.ดร.วราวุฒิ เถาลัดดา

อาจารย์ที่ปรึกษาร่วม        ดร.ศ.ทิพวรรณ คล้ายบุญมี

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง(สจล.) อนุมัติให้  
 โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต(ฟิสิกส์ประยุกต์)  
 ปีการศึกษา 2559

คณะกรรมการสอบ	ลายมือชื่อ
อ.สุรชาติ กมลดีลัก ประธานกรรมการ	
ดร.พิศาล สุขวิสูตร กรรมการ	
ดร.ภาณุพล ไชลนกระโทก กรรมการ	
รศ.ดร.วราวุฒิ เถาลัดดา กรรมการและอาจารย์ที่ปรึกษา	
ดร.ศ.ทิพวรรณ คล้ายบุญมี กรรมการและอาจารย์ที่ปรึกษาร่วม	

ลิขสิทธิ์ของคณะวิทยาศาสตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงการพิเศษ	การปรับปรุงระบบการจุ่มเคลือบควบคุมด้วยไมโครคอนโทรลเลอร์
ชื่อนักศึกษา	นางสาวศิริพร ทองเชื้อ รหัสนักศึกษา 56051220 นางสาวอุไรพร วิหกหงษ์ รหัสนักศึกษา 56051254
ปริญญา	วิทยาศาสตร์บัณฑิต (ฟิสิกส์ประยุกต์)
ภาควิชา	ฟิสิกส์
คณะ	วิทยาศาสตร์
มหาวิทยาลัย	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
ปีการศึกษา	2559
อาจารย์ที่ปรึกษา	รศ.ดร.วราวุฒิ เถาถัดตา
อาจารย์ที่ปรึกษาร่วม	ดร.ศ.ทิพวรรณ คล้ายบุญมี

### บทคัดย่อ

โครงการพิเศษนี้เป็นการปรับปรุงระบบเคลือบฟิล์มบางโดยวิธีจุ่มเคลือบ ระบบเดิมประกอบด้วยส่วนประกอบหลัก 3 ส่วน ได้แก่ (1) ส่วนขับเคลื่อนชิ้นงาน (2) ส่วนขับเคลื่อนมอเตอร์ 5 เฟส และ (3) ไมโครคอนโทรลเลอร์พร้อมซอฟต์แวร์ควบคุมการทำงาน ส่วนขับเคลื่อนชิ้นงานยังคงประกอบด้วยอุปกรณ์จับยึดชิ้นงานซึ่งติดตั้งอยู่บนแท่นขับเคลื่อนเชิงเส้น ซึ่งถูกทำให้เคลื่อนที่ในแนวตั้งด้วยสเตปปีงมอเตอร์ชนิด 5 เฟส ส่วนที่ได้รับการปรับปรุงได้แก่ ส่วนขับเคลื่อนมอเตอร์ ได้เปลี่ยนมาใช้ส่วนขับเคลื่อนสเตปปีงมอเตอร์ของ VEXTA รุ่น UDK5107N ส่วนระบบควบคุมเปลี่ยนมาใช้บอร์ด Arduino Uno R3 ซึ่งเป็นไมโครคอนโทรลเลอร์ตระกูล AVR มีการรับคำสั่งจากผู้ใช้งานผ่านทาง Rotary Encoder แทนการใช้คีย์แพดและแสดงผลทางจอแอลซีดีแบบกราฟิก (GLCD) ความละเอียด 128 x 64 แทนการใช้จอแอลซีดีแบบตัวหนังสือ (Character LCD) ขนาด 16 x 4 ตัวอักษร ระบบที่ได้รับการปรับปรุงนี้สามารถควบคุมการจุ่มด้วยอัตราเร็วระหว่าง 0.01 ถึง 6.00 มิลลิเมตรต่อวินาที ด้วยระยะเคลื่อนที่สูงสุด 9 เซนติเมตร ระบบนี้ได้รับการทดสอบโดยการจุ่มเคลือบสารละลายพอลิสด์รีนบนแผ่นแก้วพบว่าฟิล์มที่ได้มีความสม่ำเสมอสำหรับการจุ่มเคลือบด้วยเงื่อนไขที่เหมาะสม

**คำสำคัญ:** Rotary encoder, Arduino UNO-R3, GLCD

<b>Title</b>	Improvement of Dip-Coating System Based on Microcontroller
<b>Students</b>	Miss Siriporn Thongchuea Student ID 56051220 Miss Uraiporn Wihokhong Student ID 56051254
<b>Degree</b>	Bachelor of Science (Applied Physics)
<b>Department</b>	Physics
<b>Faculty</b>	Science
<b>University</b>	King Mongkut's Institute of Technology Ladkrabang (KMITL)
<b>Academic Year</b>	2016
<b>Advisor</b>	Associate Professor Dr. Warawoot Thowladda
<b>Co-advisor</b>	Dr.S.Tipawan Khlayboonme

### Abstract

This special project proposes an improvement of an automated dip-coating system used for coating thin films. The coating is controlled via a stepping motor. The previous an automated system consist of four main parts: (1) a substrate-driving part, (2) a a 5-phase stepping motor driven linear translator, (3) a motor driver and motor-control software and (4) an user-interface part. For the previous system, the motor was controlled by a constructed driver via an AT89C51ED2 microcontroller. For the improved dip-coating system, the motor driver VEXTA model UDK5107N is used instead and it is controlled via a microcontroller broad Arduino Uno R3. In addition, the rotary encoder and Graphic LED 128x64 display also used instead the keypad and character LCD 16x4 for the new improved system. This new system can move the substrate with speed ranging from 0.01 to 6.00 mm/s and a maximum distance of 9 cm. Efficiency of the system was tested by coating polystyrene on glass substrates. The results showed that the films were uniform for only some coating-conditions.

**Keywords:** Rotary encoder, Arduino UNO-R3, GLCD

## กิตติกรรมประกาศ

โครงการพิเศษนี้สามารถสำเร็จลุล่วงไปได้ด้วยดีเนื่องจากได้รับการสนับสนุนช่วยเหลือและความอนุเคราะห์จากบุคคลหลายท่าน ซึ่งต้องกราบขอบพระคุณไว้ ณ โอกาสนี้ด้วย

รศ.ดร. วรารุณี เถาลัดดา ผู้ซึ่งถ่ายทอดวิชาความรู้และให้คำปรึกษาในการสร้างเครื่องมือซึ่งเป็นประโยชน์อย่างมากในการทำโครงการพิเศษนี้และยังเป็นประโยชน์ในการนำความรู้และเทคนิคต่างๆไปใช้ในหน้าที่การงานในอนาคตได้อีกด้วย อีกทั้งยังอดทนให้ออกาสคนไม่ขยันได้ทำงานต่อจนสำเร็จลุล่วงไปได้ด้วยดี จนเกิดความรู้ความสามารถในด้านใหม่ๆที่ไม่เคยสามารถทำได้มาก่อนจึงกราบขอบพระคุณไว้ ณ ที่นี้ด้วย

ดร. ศ.ทิพวรรณ คล้ายบุญมี และพีสมภพ แซ่เจี๋ย ผู้ซึ่งคอยสนับสนุนการทำโครงการพิเศษนี้ในทุกๆด้าน ทั้งด้านทักษะความรู้ความสามารถในการทำงานและสนับสนุนอุปกรณ์และเครื่องมือต่างๆในการทำงานให้เป็นอย่างดี จึงกราบขอบพระคุณไว้ ณ ที่นี้ด้วย

เพื่อนร่วมงานในห้องปฏิบัติการ และนอกห้องปฏิบัติการ ที่ให้ความช่วยเหลือระหว่างการทำงานหลายอย่าง อีกทั้งยังให้คำแนะนำเกี่ยวกับการทำงานในด้านต่างๆจึงขอกล่าวขอบคุณในน้ำใจของเพื่อนๆร่วมงานไว้ ณ ที่นี้ด้วย

สุดท้ายขอกราบขอบพระคุณบุคคลผู้ที่มีความสำคัญที่สุดในชีวิต ได้แก่ บิดา มารดา และครอบครัวที่อบอุ่น ผู้คอยอบรมเลี้ยงดูสนับสนุนการกระทำทุกอย่างในทางที่ดีอย่างเต็มที่ ในระหว่างการทำโครงการพิเศษนี้มีปัญหาต่างๆมากมายซึ่งบั่นทอนกำลังใจในการทำงานเป็นอย่างมากแต่เนื่องจากกำลังใจของท่านเหล่านี้จึงทำให้โครงการพิเศษนี้สามารถดำเนินต่อและสำเร็จลุล่วงไปได้ด้วยดี จึงขอกล่าวขอบพระคุณท่านไว้ในโอกาสนี้ด้วย

ศิริพร ทองเชื้อ  
อุไรพร วิทหงษ์

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ช
สารบัญรูป	ซ
คำย่อ/สัญลักษณ์	ญ
<b>บทที่ 1 บทนำ</b>	<b>1</b>
1.1 ความเป็นมาและความสำคัญ	1
1.2 วัตถุประสงค์ของงานวิจัย	1
1.3 ขอบเขตของงานวิจัย	1
1.4 ขั้นตอนการดำเนินงานวิจัย	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
<b>บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง</b>	<b>3</b>
2.1 การเคลือบฟิล์มบาง	3
2.1.1 วิธีทางเคมี (Chemical Process)	3
2.1.2 วิธีทางกายภาพ (Physical Process)	3
2.2 เทคนิคการจุ่มเคลือบ (Dip Coating Method)	3
2.3 สเตปปีงมอเตอร์ (Stepping Motor)	5
2.3.1 ลักษณะโครงสร้างของสเตปปีงมอเตอร์	5
2.3.1.1 แบบแม่เหล็กถาวร	5
2.3.1.2 แบบแปรผันคาร์รีล็คแทนซ์	6
2.3.1.3 แบบผสม	6
2.3.2 การพันขดลวดของสเตปปีงมอเตอร์	6
2.3.2.1 แบบไบโพลาร์(Bipolar)	6
2.3.2.2 แบบยูนิโพลาร์(Unipolar)	7
2.3.3 การควบคุมการหมุนของสเตปปีงมอเตอร์	7
2.2.3.1 แบบฟูลสเต็ป1 เฟส (Full-Step 1 Phase)	7
2.2.3.2 แบบฟูลสเต็ป2 เฟส (Full-Step 2 Phase)	8
2.2.3.3 แบบฮาล์ฟสเต็ป (Half Step)	8
2.4 Arduino UNO R3	9
2.4.1 จุดเด่นที่ทำให้บอร์ด Arduino เป็นที่นิยม	10
2.4.2 รูปแบบการเขียนโปรแกรมบน Arduino	10

## สารบัญ(ต่อ)

	หน้า
2.4.3 Layout And Pin Out Arduino Board(Model: Arduino UNO R3)	12
2.5 Rotary Encoder	13
2.6 จอแสดงผล GLCD (Graphic Liquid Crystal Display)	13
2.6.1 Text Mode	13
2.6.2 Graphic Mode และ Picture Mode	14
<b>บทที่ 3 วิธีการดำเนินงานวิจัย</b>	<b>15</b>
3.1 การออกแบบระบบฮาร์ดแวร์เครื่อง Dip-Coating	15
3.1.1 ชุดอุปกรณ์ของโครงการ	15
3.2 หลักการเชื่อมต่ออุปกรณ์	17
3.2.1 แสดงการต่อจอ GLCD	17
3.2.2 แสดงการต่อ Rotary Encoder	18
3.3 ระบบขับเคลื่อน (Driver)	19
3.3.1 Motor Running Current Adjustment Rotary Switch	20
3.3.2 Motor Standstill Current Adjustment Rotary Switch	21
3.3.3 การปรับฟังก์ชันการใช้งาน	22
3.4 การออกแบบระบบซอฟต์แวร์เครื่อง Dip-Coating	23
3.4.1 การออกแบบส่วนติดต่อกับผู้ใช้	23
3.4.2 การทำงานของระบบซอฟต์แวร์	24
3.5 โครงสร้าง Stepping Motor 5 Phase	29
3.6 แหล่งจ่ายไฟกระแสตรง	30
3.7 การแสดงผล (GLCD)	31
3.7.1 ผลการทำงานของซอฟต์แวร์ส่วนการติดต่อกับผู้ใช้	31
3.8 วัสดุเครื่องมือ	33
3.8.1 แผ่นรองรับ	33
3.8.2 สารละลาย	34
3.9 ระบบเคลือบฟิล์มแบบจุ่มเคลือบ	34
3.10 ความเร็วในการจุ่มเคลือบ	36
3.11 การทดสอบระบบการจุ่มเคลือบ	37
3.12 ขั้นตอนกระบวนการเคลือบ	38
<b>บทที่ 4 ผลการวิจัยและการอภิปรายผล</b>	<b>41</b>
4.1 ผลการทดลอง	41
4.1.1 ผลการทดลองการเคลือบฟิล์มแบบจุ่มเคลือบ	41

## สารบัญ(ต่อ)

	หน้า
4.1.2 การเปรียบเทียบในกาจุ่มเคลือบแต่ละรอบ	42
4.1.2.1 การจุ่มเคลือบฟิล์มจำนวน 1 รอบ	42
ความเร็วในการตั้งขึ้นแตกต่างกัน	
4.1.2.2 การจุ่มเคลือบฟิล์มจำนวน 3 รอบ	42
ความเร็วในการตั้งขึ้นแตกต่างกัน	
4.2 การอภิปรายผล	43
<b>บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ</b>	<b>44</b>
5.1 สรุปผลการวิจัย	44
เอกสารอ้างอิง	
ภาคผนวก	



## สารบัญตาราง

ตารางที่	หน้า
ตารางที่ 1.1 ขั้นตอนการดำเนินงาน	2
ตารางที่ 2.1 แสดงการกระตุ้นขดลวดสเตปป์มอเตอร์แบบฟูลสเตป 1 เฟส	7
ตารางที่ 2.1 แสดงการกระตุ้นขดลวดสเตปป์มอเตอร์แบบฟูลสเตป 1 เฟส	8
ตารางที่ 2.2 แสดงการกระตุ้นขดลวดสเตปป์มอเตอร์แบบฟูลสเตป 2 เฟส	8
ตารางที่ 2.3 แสดงการกระตุ้นขดลวดสเตปป์มอเตอร์แบบฮาล์ฟสเตป	9
ตารางที่ 3.1 แสดงฟังก์ชันของจอ GLCD	17
ตารางที่ 3.2 แสดงฟังก์ชันการใช้งานของ Rotary Encoder	18
ตารางที่ 3.3 แสดงวิธีการขับ Stepping Motor ชนิด 5 เฟส	29
ตารางที่ 3.4 ความสัมพันธ์ของอัตราเร็วกับความถี่	36



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป

รูปที่	หน้า
รูปที่ 2.1 กระบวนการจุ่มเคลือบ	4
รูปที่ 2.2 ขั้นตอนการจุ่มเคลือบแบบกะ	4
รูปที่ 2.3 5 Phase stepping Motor	5
รูปที่ 2.4 แสดงลักษณะการพันขดลวดแบบไบโพลาร์ (Bipolar)	6
รูปที่ 2.5 แสดงลักษณะการพันขดลวดแบบยูนิโพลาร์ (Unipolar)	7
รูปที่ 2.6 บอร์ดArduino ต่อกับอุปกรณ์ภายนอก	9
รูปที่ 2.7 เขียนโปรแกรมบนคอมพิวเตอร์ ผ่านทางโปรแกรม Arduino IDE	10
รูปที่ 2.8 เลือกหุ่นบอร์ด Arduino ที่ต้องการ upload	10
รูปที่ 2.9 เลือกหมายเลข Comport ของบอร์ด	11
รูปที่ 2.10 Verify เพื่อตรวจสอบความถูกต้องและ Compile โค้ดโปรแกรม	11
รูปที่ 2.11 ส่วนประกอบของ Arduino UNO R3	12
รูปที่ 2.12 Rotary Encoder	13
รูปที่ 2.13 จอGLCD 128x64	14
รูปที่ 3.1 ส่วนควบคุมการทำงาน(Arduino UNO R3)	15
รูปที่ 3.2 ส่วนควบคุมระบบขับเคลื่อน(Driver)	15
รูปที่ 3.3 แหล่งจ่ายไฟให้กับ DC Power Supply (Transformer)	16
รูปที่ 3.4 ส่วนควบคุมการสั่งงาน (Rotary Encoder)	16
รูปที่ 3.5 ส่วนการแสดงผล(GLCD)	16
รูปที่ 3.6 การเชื่อมต่อจอGLCD กับ Arduino	17
รูปที่ 3.7 ไดอะแกรมRotary Encoder	18
รูปที่ 3.8 การเชื่อมต่อจอ Rotary Encoderกับ Arduino	18
รูปที่ 3.9 ชุดขับเคลื่อนมอเตอร์ (Motor Driver)	19
รูปที่ 3.10 โหมดการใช้งานในการจ่ายกระแสให้กับ (Driver)	20
รูปที่ 3.11 การจ่ายกระแสให้กับ Driver	20
รูปที่ 3.12 โหมดการใช้งานในการจ่ายกระแสให้กับ (Motor Driver)	21
รูปที่ 3.13 เปรอ์เซ็นต์การจ่ายกระแสให้กับ UDK5107N	21
รูปที่ 3.14 ฟังก์ชันการใช้งานในการทดสอบ	22
รูปที่ 3.15 ฟังก์ชันการใช้งานในการทดสอบแต่ละโหมด	22
รูปที่ 3.16 การเชื่อมต่อจอ GLCD, Rotary Encoder กับ Arduino	23
รูปที่ 3.17 ตัวอย่างการแสดงผลทางจอGLCD ของระบบสังเคราะห์ฟิล์มแบบจุ่มเคลือบ	24
รูปที่ 3.18 ไดอะแกรมส่วนควบคุมการสั่งงาน	25
รูปที่ 3.19 ไดอะแกรมการทำงานระบบสังเคราะห์ฟิล์มแบบจุ่มเคลือบ	26
รูปที่ 3.20 การเคลื่อนที่ของมอเตอร์ขณะจุ่มลง	27

## สารบัญรูป(ต่อ)

รูปที่	หน้า
รูปที่ 3.21 การเคลื่อนที่ของมอเตอร์ขณะตั้งขึ้น	28
รูปที่ 3.22 โครงสร้าง 5 Phase Stepping Motor	29
รูปที่ 3.23 แหล่งจ่ายไฟฟ้ากระแสสลับ 110V	30
รูปที่ 3.24 แหล่งจ่ายไฟฟ้ากระแสตรง 12-16V	30
รูปที่ 3.25 การแสดงผลทางจอ GLCD 128x64	33
รูปที่ 3.26 ขั้นตอนการเตรียมสารละลาย Polystyrene	34
รูปที่ 3.27 โครงสร้างภายในของระบบเคลือบแบบจุ่ม	35
รูปที่ 3.28 ระบบเคลือบฟิล์มแบบจุ่มเคลือบ	35
รูปที่ 3.29 ตัวอย่างการวัดค่าจากออสซิลโลสโคปที่ความเร็ว Speed 1 mm/s	37
รูปที่ 3.30 แผนภาพขั้นตอนกระบวนการเคลือบฟิล์มในการทดลองที่ 1	38
รูปที่ 3.31 แผนภาพขั้นตอนกระบวนการเคลือบฟิล์มในการทดลองที่ 2	39
รูปที่ 3.32 แผนภาพขั้นตอนกระบวนการเคลือบฟิล์มในการทดลองที่ 3	40

## คำย่อ/สัญลักษณ์

คำย่อ/สัญลักษณ์	คำอธิบาย
GLCD	Graphic Liquid Crystal Display



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญ

การเคลือบผิววัสดุมีวัตถุประสงค์เพื่อปรับปรุงสมบัติของวัสดุให้เป็นไปตามที่ต้องการ ทั้งนี้การปรับปรุงสมบัติของวัสดุให้เหมาะสมกับการใช้งาน เช่น การเคลือบพื้นผิววัสดุเพื่อทำให้เกิดเป็นชั้นฟิล์มบางบนฐานรอง กระบวนการและวิธีการเคลือบฟิล์มบางที่มีต้นทุนต่ำ ได้แก่ การเคลือบชั้นฟิล์มบางด้วยการจุ่มเคลือบ

โครงการพิเศษนี้ เป็นการปรับปรุงระบบสังเคราะห์ฟิล์มบางโดยวิธีจุ่มเคลือบที่มีอยู่เดิมแล้ว ซึ่งระบบนี้ชิ้นงานจะถูกจุ่มและดึงขึ้นจากสารละลายพอลิเมอร์ หรือเรียกว่า กระบวนการเกิดพอลิเมอร์ เพราะเป็นเทคนิคที่ใช้อุณหภูมิต่ำและสามารถเตรียมฟิล์มได้ทั่วทั้งพื้นที่ผิวที่มีขนาดใหญ่ เช่น การเคลือบกระจกขนาดใหญ่ในอุตสาหกรรม เป็นต้น นอกจากนี้เทคนิคนี้ยังใช้ต้นทุนต่ำทำได้ง่ายและไม่ต้องใช้เครื่องมือที่ซับซ้อน สมบัติของฟิล์มและอนุภาคระดับนาโนที่ได้จะขึ้นกับปัจจัยในการเตรียมหลายอย่างเช่น อัตราการจุ่มเคลือบ ความเข้มข้นของสารตั้งต้นและชนิดของตัวทำละลายที่ใช้ ระบบนี้จะใช้สเตปปิงมอเตอร์ (stepping motor) ในการควบคุมอัตราเร็วของการเคลื่อนที่ของชิ้นงานในการจุ่มลงหรือยกชิ้นงานขึ้นจากสารละลาย ซึ่งมอเตอร์จะถูกควบคุมด้วยไมโครคอนโทรลเลอร์

### 1.2 วัตถุประสงค์ของงานวิจัย

1. ศึกษาการทำงานและการควบคุมสเตปปิงมอเตอร์ชนิด 5 เฟส
2. เพื่อศึกษาการทำงานของชุดขับเคลื่อนสเตปปิงมอเตอร์ (Motor Driver) ชนิด 5 เฟส
3. เพื่อศึกษาการสร้างระบบสังเคราะห์ฟิล์มบางโดยวิธีการจุ่มเคลือบ
4. เพื่อศึกษาการทำงานและการใช้ไมโครคอนโทรลเลอร์ Arduino UNO R3 ในการควบคุมและแสดงผล
5. เพื่อปรับปรุงระบบป้อนข้อมูลและแสดงผลจากการป้อนข้อมูลผ่านคีย์แพดและแสดงผลทางจอแอลซีดีแบบตัวอักษรในระบบเก่ามาเป็นการป้อนข้อมูลผ่าน Rotary Encoder และแสดงผลทางจอแอลซีดีแบบกราฟิก

### 1.3 ขอบเขตของงานวิจัย

ขอบเขตของงานวิจัยสามารถแบ่งเป็นขั้นตอนได้ดังนี้

1. ศึกษาการทำงานและการควบคุมสเตปปิงมอเตอร์ชนิด 5 เฟส
2. ควบคุมมอเตอร์และแทนจับชิ้นงานให้เคลื่อนที่ไปได้ในระยะเวลาที่ต้องการ
3. สามารถควบคุมการทำงานของมอเตอร์โดยการป้อนข้อมูลผ่าน Rotary Encoder
4. ใช้ไมโครคอนโทรลเลอร์ Arduino UNO R3 ในการควบคุมและการแสดงผลผ่านจอ GLCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.4 ขั้นตอนการดำเนินงานวิจัย

ในการวิจัยนี้จะทำการปรับปรุงและพัฒนาาระบบสังเคราะห์ฟิล์มในส่วนที่มีอยู่แล้ว โดยใช้ไมโครคอนโทรลเลอร์ Arduino UNO R3 ควบคุมระบบการทำงาน ใช้ Rotary Encode เป็นภาคอินพุตแทนการใช้คีย์แพดและใช้จอแสดงผลแอลซีดีแบบกราฟิก (Graphic Liquid Crystal Display) ความละเอียด 128 x 64 เป็นภาคเอาต์พุตแทนการใช้จอแอลซีดีแบบตัวอักษร (Character LCD) ขนาด 16 x 4 ตัวอักษร โดยมีขั้นตอนการดำเนินงานดังตารางที่ 1.1

ตารางที่ 1.1 ขั้นตอนการดำเนินงาน

แผนการดำเนินงาน	พ.ย	ธ.ค	ม.ค	ก.พ	มี.ค	เม.ย	พ.ค	มิ.ย
1. ศึกษาค้นคว้าทฤษฎีระบบสังเคราะห์ฟิล์มบางโดยวิธีการจุ่มเคลือบ								
2. ศึกษาการสั่งการของ Rotary encoder ผ่านจอ GLCD และใช้ Arduino UNO R3 ในการควบคุม								
3. ปรับปรุงระบบการสังเคราะห์ฟิล์มบางโดยวิธีการจุ่มเคลือบ								
4. ทดสอบฟิล์มด้วยระบบการสังเคราะห์ฟิล์มบางโดยวิธีการจุ่มเคลือบที่ปรับปรุงขึ้น								
5. เขียนเล่มโครงการพิเศษ								

## 1.5 ผลที่คาดว่าจะได้รับ

1. ทำให้ทราบถึงระบบการจุ่มเคลือบฟิล์มบางโดยวิธีการจุ่มเคลือบ โดยใช้ไมโครคอนโทรลเลอร์ Arduino UNO R3 ควบคุมระบบการทำงาน
2. สามารถนำความรู้ที่ได้จากการปรับปรุงระบบสังเคราะห์ฟิล์มบางโดยใช้ไมโครคอนโทรลเลอร์ไปต่อยอดกับทางด้านอื่นๆได้อย่างมีประสิทธิภาพ

## บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

### 2.1 การเคลือบฟิล์มบาง [2]

ฟิล์มบาง (thin film) หมายถึง ชั้นของอะตอมหรือกลุ่มของอะตอมที่จับรวมกันเป็นชั้นบางๆ การระบุว่าฟิล์มใดเป็น "ฟิล์มบาง" อาจพิจารณาได้จากลักษณะการใช้งานว่าใช้สมบัติด้านใดของฟิล์ม กล่าวคือ ถ้าเป็นการใช้สมบัติเชิงผิว (surface properties) จะเรียกฟิล์มนั้นว่า "ฟิล์มบาง" แต่ถ้าเป็นการใช้สมบัติเชิงปริมาตร (bulk properties) จะเรียกฟิล์มนั้นว่า "ฟิล์มหนา" ทั้งนี้จากความหมายจะเห็นว่าฟิล์มเดียวกันอาจเป็น "ฟิล์มบาง" หรือ "ฟิล์มหนา" ก็ได้ขึ้นกับลักษณะการใช้งานเป็นสำคัญ อย่างไรก็ตาม การเคลือบฟิล์มบางเป็นกระบวนการที่มีวัตถุประสงค์เพื่อเปลี่ยนแปลงหรือปรับปรุงสมบัติของวัสดุที่เคลือบให้ดีขึ้น เช่น ให้ความคงทน ทนทาน หรือ สวยงามมากขึ้น รวมถึงเพื่อการใช้งานประโยชน์ในงานอื่นๆ การสร้างทำได้โดยการเคลือบฟิล์มบางจะเป็นกระบวนการทำให้สารเคลือบตกเคลือบบนผิวของวัสดุที่ต้องการ โดยเริ่มจากการสร้างสารเคลือบในรูปของอะตอม โมเลกุล หรืออนุภาคจากนั้นเป็นการเคลื่อนย้ายสารเคลือบมายังวัสดุรองรับเมื่อสารเคลือบตกกระทบบนผิววัสดุรองรับก็เกิดการควบแน่นและพอกพูนเป็นชั้นของฟิล์มบางบนวัสดุรองรับในที่สุด

วิธีการเตรียมฟิล์มบางมีหลายแบบทั้งนี้ขึ้นอยู่กับเทคโนโลยี ปริมาณ และคุณภาพของฟิล์มบาง สำหรับเทคโนโลยีในการเตรียมฟิล์มบางนั้นแบ่งได้ 2 วิธีคือ

#### 2.1.1 วิธีทางเคมี (Chemical Process)

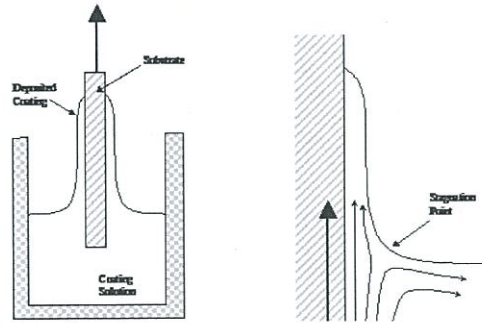
เป็นการเคลือบที่อาศัยการแตกตัวของสารเคมีในสภาพของก๊าซ และเกิดปฏิกิริยาเคมีกลายเป็นสารใหม่ตกเคลือบบนแผ่นรองรับ ซึ่งรวมถึงการพ่นสเปรย์ไพโรไลซิส (Spray Pyrolysis) กระบวนการเคลือบไอเคมี (Chemical vapor deposition) และวิธีโซลเจล (Sol-gel)

#### 2.1.2 วิธีทางกายภาพ (Physical Process)

เป็นการเคลือบที่อาศัยการทำให้อะตอมของสารเคลือบหลุดออกจากผิวแล้วฟุ้งกระจายหรือวิ่งเข้าไปจับและยึดติดกับผิวของแผ่นรองรับ โดยการใช้ความร้อน (Thermal) และระเหยสารด้วยลำอิเล็กตรอน (Electron beam evaporation) รวมถึงวิธีการใช้แสงเลเซอร์ (Laser ablation) และวิธีสปัตเตอริง (Sputtering)

### 2.2 เทคนิคการจุ่มเคลือบ (Dip Coating Method) [8]

ในกระบวนการจุ่มเคลือบแผ่นรองรับจะถูกจุ่มลงไปในสารละลายแล้วถูกดึงขึ้นจากสารละลาย โดยการควบคุมความเร็ว โดยทั่วไปแล้วความหนาในการเคลือบจะเพิ่มขึ้นโดยแผ่นรองรับ จะถูกดึงในความเร็วที่เร็วขึ้น ความหนาของฟิล์มจะถูกกำหนดโดยความสมดุลของแรงที่จุดเฉื่อยชา หรือจุดหยุดนิ่ง (Stagnation) บนผิวของเหลว ความเร็วในการดึงที่เร็วขึ้นผิวของแผ่นรองรับจะดึงของเหลวได้มากขึ้น ก่อนที่จะไหลกลับลงไปในสารละลาย ความหนาที่ได้จะเป็นผลมาจากความหนืดของของเหลว ความหนาแน่นของของเหลว แรงตึงผิว และความเร็วในการดึง

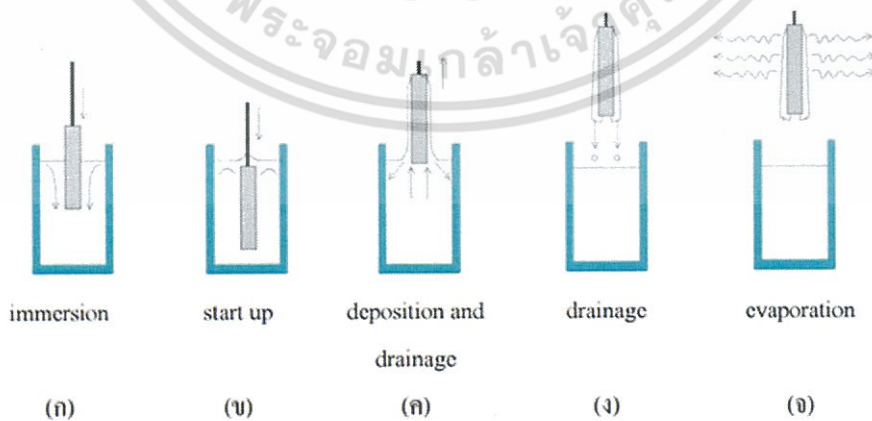


รูปที่ 2.1 กระบวนการจุ่มเคลือบ [12]

ในการเคลือบแบบจุ่มนั้น เพื่อให้ได้ผลิตภัณฑ์ที่มีคุณภาพสูง หรือมีชุดในการเคลือบที่ดีจะต้อง มีการควบคุมกระบวนการที่แม่นยำและมีสภาพแวดล้อมที่สะอาดอีกด้วย ในการเคลือบแผ่นรองรับ อาจเปียกอยู่หลายนาที่จนกว่าไอของตัวทำละลายจะระเหยออกไป ในกระบวนการนี้สามารถเร่งให้แห้ง โดยให้ความร้อน UV, หรือเทคนิคของอินฟราเรด (IR) แต่ก็ขึ้นอยู่กับสารละลายที่ใช้ในการเคลือบ วิธีการจุ่มเคลือบผิวที่นิยมใช้กันโดยทั่วไป คือ การจุ่มเคลือบแบบกะ (Batch Dip Coating)

การจุ่มเคลือบแบบกะ (Batch Dip Coating) ในการจุ่มเคลือบแบบกะนั้นสามารถแบ่งขั้นตอน ในขณะเคลือบได้ 5 ขั้นตอน ดังแสดงในรูปที่ 2.2(ก) ถึง (จ) ดังนี้

- (ก) ขั้นตอนการจุ่มชิ้นงาน (Immersion)
- (ข) ขั้นตอนการเริ่มต้นดึงชิ้นงานขึ้น (Start up)
- (ค) ขั้นตอนที่สารละลายเกาะที่ผิวชิ้นงานเริ่มไหลย้อนกลับ (Deposition & Drainage)
- (ง) สารละลายไหลแยกออกจากชิ้นงาน (Drainage)
- (จ) ขั้นตอนที่สารละลายระเหย (Evaporation) สำหรับตัวทำละลายที่ระเหยง่าย เช่น แอลกอฮอล์ ขั้นตอนการระเหยมักจะเกิดขึ้นพร้อมๆกับขั้นตอน (ข) ถึง (จ)



รูปที่ 2.2 ขั้นตอนการจุ่มเคลือบแบบกะ[13]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.3 สเตปปีงมอเตอร์ (Stepping Motor) [1]

สเตปปีงมอเตอร์ เป็นมอเตอร์แบบหนึ่งที่มีการหมุนแบบเป็นสเตปกล่าวคือ เมื่อจ่ายสัญญาณให้แก่มอเตอร์อย่างถูกต้อง มอเตอร์จะหมุนไปเป็นจังหวะไม่เหมือนกับมอเตอร์ในแบบทั่วไปที่เมื่อทำการจ่ายไฟแล้วหมุนทันที หมุนตามเข็มหรือทวนเข็มขึ้นอยู่กับขั้วของการจ่ายไฟให้แก่ขดลวดภายในมอเตอร์ ความละเอียดของการหมุนของสเตปปีงมอเตอร์นี้จะขึ้นอยู่กับองค์ประกอบสำคัญ 2 ประการคือ โครงสร้างของสเตปปีงมอเตอร์ และวงจรขับ โดยวงจรขับช่วยให้การหมุนของสเตปปีงมอเตอร์มีความละเอียดมากขึ้นกว่าความละเอียดของการเคลื่อนที่ในแต่ละสเตปได้ โดยสเตปมอเตอร์เป็นอุปกรณ์เชิงกลอีกชนิดหนึ่งที่มีการนำเอามาประยุกต์ใช้งานกันในหลายๆด้านอย่างแพร่หลาย โดยเฉพาะในด้านของงานอุตสาหกรรมที่เกี่ยวกับเครื่องกลการผลิตสมัยใหม่ รวมถึงหุ่นยนต์ที่ใช้ในงานอุตสาหกรรม ซึ่งส่วนมากมักจะใช้สเตปปีงมอเตอร์เป็นส่วนประกอบอยู่ด้วยเสมอ



รูปที่ 2.3 5 Phase Stepping Motor

### 2.3.1 ลักษณะโครงสร้างของสเตปปีงมอเตอร์

สเตปปีงมอเตอร์สามารถจำแนกออกได้ตามลักษณะโครงสร้างได้ 3 แบบต่อไปนี้

#### 2.3.1.1 แบบแม่เหล็กถาวร (Permanent Magnet Step Motor หรือ PM)

สเตปปีงมอเตอร์แบบ PM นี้จะใช้ส่วนของสเตเตอร์ (Stator) สำหรับพันขดลวดไว้หลายๆโพล (Pole) และจะมีส่วนโรเตอร์ (Rotor) เป็นรูปทรงกระบอกฟันเลื่อย ซึ่งส่วนของโรเตอร์นี้ทำด้วยแม่เหล็กถาวร โดยเมื่อเราป้อนไฟฟ้ากระแสตรงให้กับขดลวดที่พันอยู่บนส่วนของแกนสเตเตอร์ทำให้เกิดสนามแม่เหล็กไฟฟ้าขึ้นในขดลวดสเตเตอร์และสนามแม่เหล็กไฟฟ้าที่เกิดขึ้นนี้จะทำให้เกิดแรงดันหรือผลกับขั้วแม่เหล็กของส่วนโรเตอร์จึงส่งผลให้แกนโรเตอร์ของมอเตอร์สามารถเคลื่อนที่ไปได้ซึ่ง สเตปปีงมอเตอร์แบบ PM นี้จะมีแรงดูดยึดส่วนของโรเตอร์หยุดอยู่กับที่ได้ดี ถึงแม้ว่ายังไม่ได้จ่ายไฟให้กับขดลวดของมอเตอร์เลยก็ตาม ลักษณะเด่นของสเตปปีงมอเตอร์แบบนี้คือ ให้แรงบิดสูงแต่ มักมีความเร็วและระยะในการเคลื่อนที่ไม่ละเอียดมากนัก

### 2.3.1.2 แบบแปรผันค่ารีลักแตนซ์ (Variable Reluctance Step Motor)

สเตปปีงมอเตอร์แบบ VR นี้ส่วนของโรเตอร์สามารถเคลื่อนที่ได้อย่างอิสระ โดยโรเตอร์ทำขึ้นจากสารแม่เหล็กกำลังอ่อน (Force Low Magnetic) มีลักษณะเป็นทรงกระบอกพื้นเลื้อย ซึ่งมีความสัมพันธ์โดยตรงกับจำนวนโพลของขดลวดในสเตเตอร์เพื่อใช้สำหรับกำหนดระยะของมุมที่หมุนไปในแต่ละครั้งของการเคลื่อนที่ เมื่อเราป้อนกระแสไฟฟ้าให้กับขดลวดในสเตเตอร์จะทำให้เกิดแรงบิดเพื่อไปหมุนโรเตอร์ให้เคลื่อนที่ไปในเส้นทางของอำนาจแม่เหล็กที่มีค่า Reluctance ต่ำที่สุดจึงทำให้มอเตอร์แบบนี้มีความเร็วสูงและมีตำแหน่งของการเคลื่อนที่ที่มีความแม่นยำสูงและมีเสถียรภาพดี แต่มีแรงบิดน้อยกว่าแบบ PM

### 2.3.1.3 แบบผสม (Hybrid Stepping Motor หรือ H)

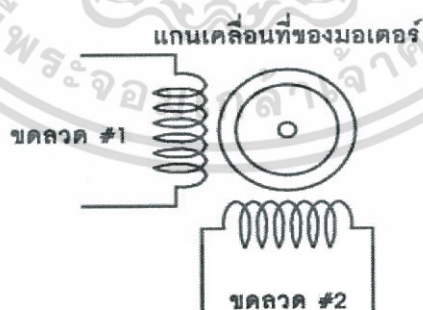
สเตปปีงมอเตอร์แบบนี้เป็นการนำเอาข้อดีของมอเตอร์แบบ PM และ VR มารวมเข้าด้วยกัน โดยจะออกแบบให้มีส่วนของสเตเตอร์มีลักษณะคล้ายกับมอเตอร์แบบ VR สำหรับในส่วนของโรเตอร์จะมีหมวกหุ้มปลายทำด้วยสารแม่เหล็กกำลังสูง โดยในการออกแบบจะควบคุมขนาดรูปร่างของหมวกแม่เหล็กเป็นอย่างดีเพื่อให้ได้มุมการหมุนของมอเตอร์ในแต่ละครั้งที่ละเอียดแม่นยำมาก มีแรงดูดยึดให้แก่โรเตอร์หยุดนิ่งอยู่กับที่ได้ดี ถึงแม้ว่าจะยังไม่ได้จ่ายกระแสไฟฟ้าให้กับขดลวดของมอเตอร์เลยก็ตามแต่มอเตอร์แบบนี้มักมีราคาสูงกว่ามอเตอร์แบบอื่นๆ

## 2.3.2 การพันขดลวดของสเตปปีงมอเตอร์

สเตปปีงมอเตอร์โดยทั่วไปมีการพันขดลวด 2 ชนิด คือ

### 2.3.2.1 แบบไบโพลาร์ (Bipolar)

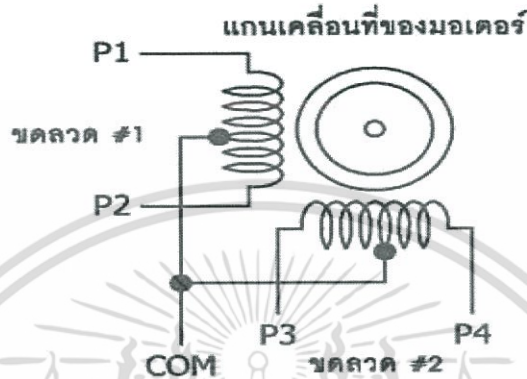
สเตปปีงมอเตอร์ชนิดนี้มีการพันขดลวด 1 ขด บนขั้วแม่เหล็กของสเตเตอร์ ขั้วแม่เหล็กของ สเตเตอร์จะถูกกำหนดโดยทิศทางการไหลของกระแสไฟฟ้าซึ่งการทำให้ขั้วแม่เหล็กเกิดทิศทางการตรงกันข้ามโดยการกลับทิศทางการไหลของกระแส โดยทั่วไปแล้วแบบไบโพลาร์จะทำให้เกิดแรงบิด (Torque) มากกว่าแบบยูนิโพลาร์



รูปที่ 2.4 แสดงลักษณะการพันขดลวดแบบไบโพลาร์ (Bipolar) [1]

### 2.3.2.2 แบบยูนิโพลาร์ (Unipolar)

สเตปป์มอเตอร์ชนิดนี้มีการพันขดลวด 2 ขด บนแต่ละขั้วแม่เหล็กของสเตเตอร์ แต่ละขดจะแบ่งออกเป็นสองเฟส รวมทั้งหมดมี 4 เฟส คือ 1,2,3 และ 4 โดยทั่วไปแล้วสเตปป์มอเตอร์ชนิดนี้จะมีแบบ 5 สาย 4 เฟส และ 6 สาย 4 เฟส โดยแบบ 5 สาย 4 เฟส จะมีสายออกมาจากมอเตอร์ 6 เส้น โดยสาย 4 เส้นแรกจะต่อที่เฟสของมอเตอร์ และสายอีก 2 เส้น (Common) จะนำมาต่อรวมกัน



รูปที่ 2.5 แสดงลักษณะการพันขดลวดแบบยูนิโพลาร์ (Unipolar) [1]

### 2.3.3 การควบคุมการหมุนของสเตปป์มอเตอร์

การให้สเตปป์มอเตอร์หมุนได้นั้นจะต้องมีการป้อนสัญญาณพัลส์ที่ขดลวดแต่ละเฟสของ สเตปป์มอเตอร์ให้ไปทิศทางเดียวกันหรือเรียงลำดับ (Sequence) และถ้าต้องการให้สเตปป์มอเตอร์หมุนกลับทิศทางก็จะต้องป้อนสัญญาณพัลส์ในทิศทางตรงกันข้าม การควบคุมการหมุนของสเตปป์มอเตอร์ สามารถแบ่งออกเป็น 3 ลักษณะดังนี้

#### 2.3.3.1 แบบฟูลสเตป 1 เฟส (Full-Step 1 Phase)

การควบคุมลักษณะนี้จะเป็นการกระตุ้นขดลวดที่ละขดเรียงตามลำดับ 1,2,3,4,5 การกระตุ้นจะมีขดลวดขดเดียวในเวลาหนึ่งที่ถูกระตุ้นเท่านั้น เช่น ขดที่ 1,2,3,4,5 หรือถ้าต้องการให้หมุนสวนทิศทางกันก็จะกระตุ้นขดลวดที่ 5,4,3,2,1 เป็นต้น การกระตุ้นแบบนี้จะทำงานง่ายที่สุดและกินกระแสไฟฟ้าน้อยที่สุดด้วยแสดงดังในตารางที่ 2.1

ตารางที่ 2.1 แสดงการกระตุ้นขดลวดสเตปป์มอเตอร์แบบฟูลสเตป 1 เฟส

สเตปป์ที่	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4	เฟสที่ 5
1	1	0	0	0	0
2	0	1	0	0	0
3	0	0	1	0	0
4	0	0	0	1	0
5	0	0	0	0	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.3.2 แบบฟูลสเตป 2 เฟส (Full-Step 2 Phase)

การควบคุมลักษณะนี้เป็นการกระตุ้นซึ่งคล้ายกับแบบฟูลสเตป 1 เฟส แต่การกระตุ้นแบบนี้จ่ายกำลังไฟฟ้าไปที่ขดลวด 2 ขดที่อยู่ใกล้กันในเวลาเดียวกัน และเรียงถัดกันไป เช่นเดียวกับแบบฟูลสเตป 1 เฟส ดังตัวอย่าง ขดลวดชุดแรกที่ถูกกระตุ้นจะเป็นขดที่ 1 และ 2 ตามด้วยการกระตุ้นขดที่ 2 และ 3 ต่อไปเป็นขดที่ 3 และ 4 และเป็นขดที่ 4 และ 5 ถัดไปเป็นขดที่ 5 และ 1 แล้วกลับมาที่ขดที่ 1 และ 2 วนไปตามลำดับเช่นนี้ หรือเริ่มที่ขด 1 และ 5 ตามด้วยขดที่ 5 และ 4 ถัดไปเป็นขดที่ 4 และ 3 ต่อไปเป็นขดที่ 3 และ 2 และเป็นขดที่ 2 และ 1 แล้ววนกลับมาที่ขดที่ 1 และ 5 ทิศทางการหมุนจะสวนทางกันการกระตุ้นสเตปเตอร์มอเตอร์แบบนี้สามารถเพิ่มแรงบิดได้มากกว่าแบบฟูลสเตป 1 เฟส โรเตอร์จะเคลื่อนที่ด้วยแรงดึงอย่างเต็มแรงจาก 2 ขดลวดที่ถูกกระตุ้นพร้อมกัน และเคลื่อนที่ต่อไปด้วยแรงดึงจากอีก 2 ขดลวดถัดไป สำหรับข้อเสียคือการกระตุ้นแบบนี้ต้องใช้กำลังไฟฟ้ามากขึ้น ขั้นตอนการทำงานต่างๆ แสดงดังในตารางตารางที่ 2.2

ตารางที่ 2.2 แสดงการกระตุ้นขดลวดสเตปปีงมอเตอร์แบบฟูลสเตป 2 เฟส

สเตปที่	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4	เฟสที่ 5
1	1	1	0	0	0
2	0	1	1	0	0
3	0	0	1	1	0
4	0	0	0	1	1
5	1	0	0	0	1

### 2.3.3.3 แบบฮาล์ฟสเตป (Half Step)

การควบคุมลักษณะนี้เป็นรูปแบบที่ผสมผสานระหว่างการกระตุ้นแบบฟูลสเตป 1 และ 2 เฟส เพื่อเพิ่มจำนวนของสเตปต่อรอบอีกเท่าตัวหนึ่ง ในระบบนี้จะกระตุ้นขดลวดเรียงกันไปเป็นลำดับดังนี้ เริ่มจากขดลวดที่ 1, แล้ววนกลับมาขดลวดที่ 1 ตามด้วยการกระตุ้นขดที่ 1 และ 2 ต่อไปเป็นขดที่ 2 ต่อไปเป็นขดที่ 2 และ 3 ต่อไปเป็นขดที่ 3 ต่อไปเป็นขดที่ 3 และ 4 ต่อไปเป็นขดที่ 4 ต่อไปเป็นขดที่ 4 และ 5 ต่อไปเป็นขดที่ 5 ต่อไปเป็นขดที่ 5 และ 1 แรงบิดที่ได้จากการกระตุ้นแบบนี้ จะเพิ่มมากขึ้นอีก เพราะช่วงสเตปมีระยะสั้นลง แต่ละ สเตปเกิดแรงดึงจากขดลวด 2 ขดที่ถูกกระตุ้นพร้อมกัน ความถูกต้องของตำแหน่งมีเพิ่มมากขึ้น แต่ต้องพึงระวังไว้อีกประการหนึ่งว่าเมื่อกระตุ้นให้ทำงานในรูปแบบนี้จะต้องทำการหมุนถึง 2 สเตป จึงจะได้เท่ากับระยะเท่ากับ 1 สเตปเต็มของการควบคุมใน 2 แบบแรก สำหรับแหล่งจ่ายกำลังไฟฟ้าต้องใช้ขนาดเท่ากับแบบ 2 เฟสเป็นอย่างน้อย จึงเพียงพอ ขั้นตอนการทำงานต่างๆ แสดงดังในตารางที่ 2.3

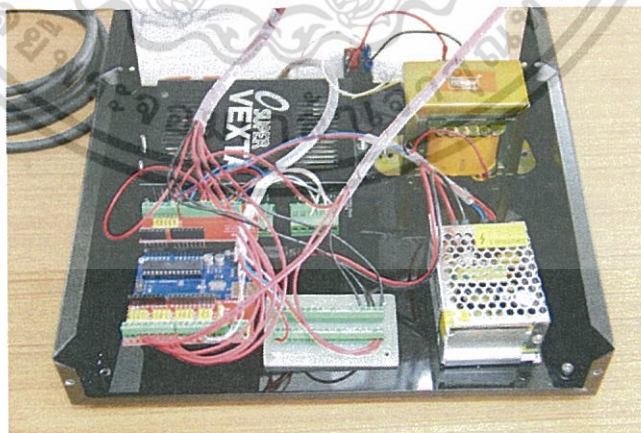
ตารางที่ 2.3 แสดงการกระตุ้นขดลวดสเตปปีงมอเตอร์แบบฮาล์ฟสเตป

สเตปที่	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4	เฟสที่ 5
1	1	0	0	0	0
2	1	1	0	0	0
3	0	1	0	0	0
4	0	1	1	0	0
5	0	0	1	0	0
6	0	0	1	1	0
7	0	0	0	1	0
8	0	0	0	1	1
9	0	0	0	0	1
10	1	0	0	0	1

## 2.4 Arduino UNO R3 [3]

Arduino อ่านว่า (อา-ดู-อิ-โน้ หรือ อาดูยโน้) เป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ดหรือโปรแกรมต่อได้อีกด้วย

ความง่ายของบอร์ด Arduino ในการต่ออุปกรณ์เสริมต่างๆ คือผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด ดังรูปที่ 2.6



รูปที่ 2.6 บอร์ด Arduino ต่อกับอุปกรณ์ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.4.1 จุดเด่นที่ทำให้บอร์ด Arduino เป็นที่นิยม

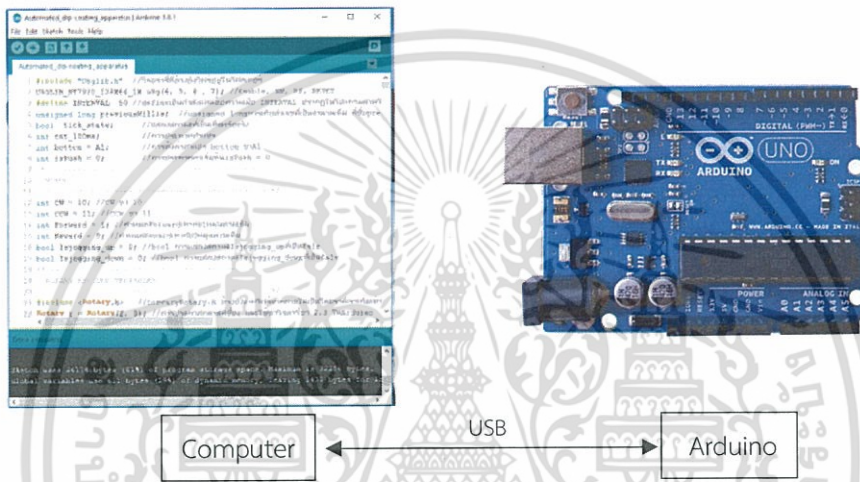
2.4.1.1 ง่ายต่อการพัฒนา มีรูปแบบคำสั่งพื้นฐาน ไม่ซับซ้อนเหมาะสำหรับผู้เริ่มต้น

2.4.1.2 มี Arduino Community กลุ่มคนที่ร่วมกันพัฒนาที่แข็งแกร่ง

2.4.1.3 Open Hardware ทำให้ผู้ใช้สามารถนำบอร์ดไปต่อยอดใช้งานได้หลายด้านราคาไม่แพง

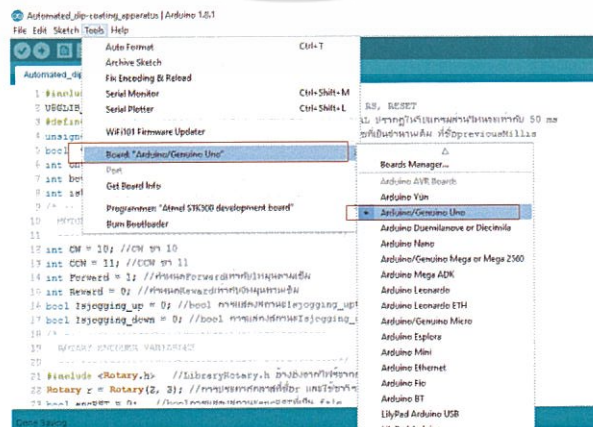
### 2.4.2 รูปแบบการเขียนโปรแกรมบน Arduino

2.4.2.1 เขียนโปรแกรมบนคอมพิวเตอร์ ผ่านทางโปรแกรม Arduino IDE ซึ่งสามารถดาวน์โหลดได้



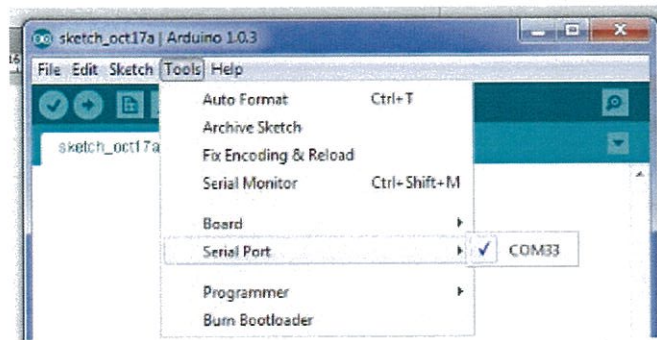
รูปที่ 2.7 เขียนโปรแกรมบนคอมพิวเตอร์ ผ่านทางโปรแกรม Arduino IDE

2.4.2.2 หลังจากที่เราเขียนโค้ดโปรแกรมเรียบร้อยแล้ว ให้ผู้ใช้งานเลือกรุ่นบอร์ด Arduino ที่ใช้ และหมายเลข Com port



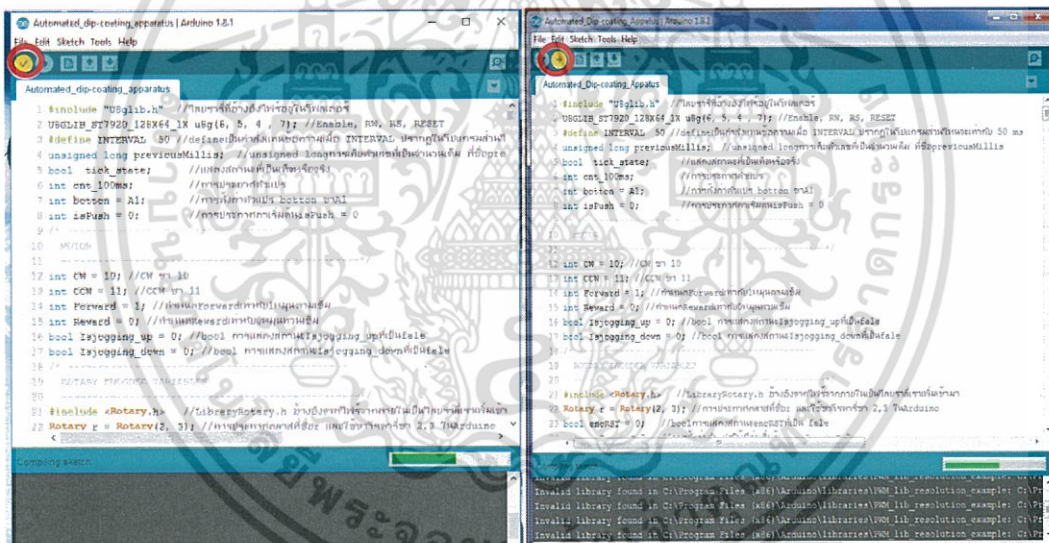
รูปที่ 2.8 เลือกุ่นบอร์ด Arduino ที่ต้องการ upload

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 เลือกหมายเลข Comport ของบอร์ด

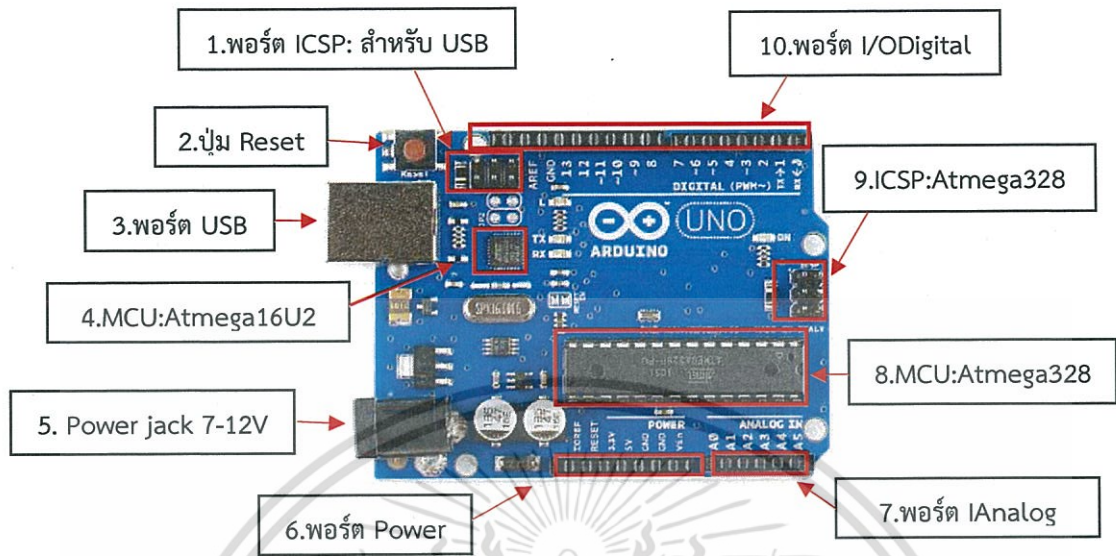
2.4.2.3 กดปุ่ม Verify เพื่อตรวจสอบความถูกต้องและ Compile โค้ดโปรแกรม จากนั้นกดปุ่ม Upload โค้ด โปรแกรมไปยังบอร์ด Arduino ผ่านทางสาย USB เมื่ออัปโหลดเรียบร้อยแล้ว จะแสดงข้อความแถบข้างล่าง “Done uploading” และบอร์ดจะเริ่มทำงานตามที่เขียนโปรแกรมไว้ได้ทันที



รูปที่ 2.10 Verify เพื่อตรวจสอบความถูกต้องและ Compile โค้ดโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.4.3 Layout And Pin Out Arduino Board (Model: Arduino UNO R3)



รูปที่ 2.11 ส่วนประกอบของ Arduino UNO R3

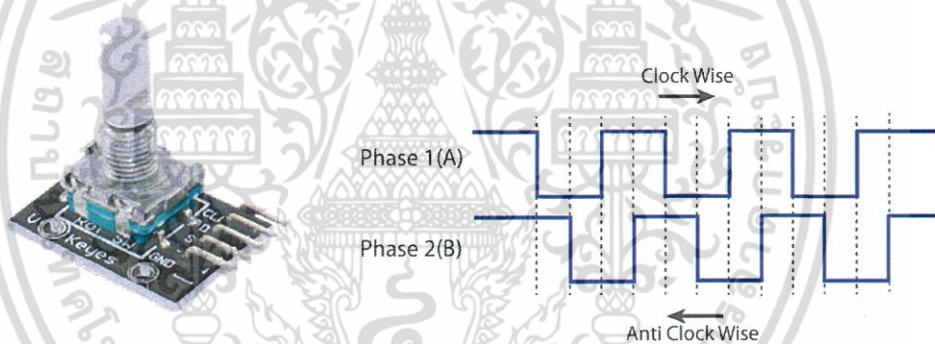
- 2.4.3.1 ICSP Port ของ Atmega16U2 เป็นพอร์ตที่ใช้โปรแกรม Visual Comport บน Atmega16U2
- 2.4.3.2 Reset Button เป็นปุ่ม Reset ใช้กดเมื่อต้องการให้ MCU เริ่มการทำงานใหม่
- 2.4.3.3 USB Port ใช้สำหรับต่อกับ Computer เพื่ออัปโหลดโปรแกรมเข้า MCU และจ่ายไฟให้กับบอร์ด
- 2.4.3.4 MCU Atmega328 เป็น MCU ที่ใช้บนบอร์ด Arduino
- 2.4.3.5 Power Jack รับไฟจาก Adapter โดยที่แรงดันอยู่ระหว่าง 7-12 V
- 2.4.3.6 Power จ่ายไฟให้กับอุปกรณ์ภายนอกที่นำมาต่อ
- 2.4.3.7 I Port Analog ช่องรับสัญญาณอนาล็อก ตั้งแต่ขา A0-A5
- 2.4.3.8 MCU ของ Atmega16U2 เป็น MCU ที่ทำหน้าที่เป็น USB to serial โดย atmega328 จะติดต่อกับ Computer ผ่าน Atmega16U2
- 2.4.3.9 ICSP PortAtmega328 เป็นพอร์ตที่ใช้โปรแกรม Boot loader
- 2.4.3.10 I/O Port Digital 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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.5 Rotary Encoder [4]

Rotary encoder ทำหน้าที่แปลงการหมุนเชิงกลให้เป็นสัญญาณทางไฟฟ้า มีรูปแบบสัญญาณเอาต์พุตที่ออกมาเป็นลักษณะของสัญญาณพัลส์ (Pulse) ที่เป็นสัญญาณรูปสี่เหลี่ยม โดยจำนวนพัลส์ (Pulse) ที่ออกมาจะมีความสัมพันธ์กับระยะการเคลื่อนที่ ตำแหน่ง ระยะห่าง ความเร็ว และความเร่ง นอกจากนี้ยังสามารถระบุได้ถึงทิศทางการหมุนตามเข็มนาฬิกาหรือทวนเข็มนาฬิกาโดยอาศัยการตรวจจับทิศทางการหมุนจากมุมเฟส (Phase) ของสัญญาณเอาต์พุต A (CLK) กับ สัญญาณเอาต์พุต B (DT) ว่าสัญญาณใดเกิดก่อน ซึ่งมีมุมเฟส (Phase) ที่ต่างกันอยู่ 90 องศา เมื่อเฟส (Phase) A เกิดขึ้นก่อนเฟส (Phase) B เป็นการหมุนตามเข็มนาฬิกา ในทางตรงกันข้ามถ้าเฟส (Phase) B เกิดขึ้นก่อนเฟส (Phase) A จะเป็นการหมุนทวนเข็มนาฬิกา

ซึ่งใช้ไมโครคอนโทรลเลอร์ Arduino UNO R3ตระกูล AVR ในการออกแบบชุดคำสั่งโปรแกรม มีการรับคำสั่งจากผู้ใช้งานผ่านทาง Rotary Encoder ในการป้อนข้อมูลสามารถหมุนทวนเข็มนาฬิกาหรือตามเข็มนาฬิกาได้ตามต้องการ และสามารถกดเลือกแบบสวิตซ์ได้ซึ่งข้อมูลที่เรอป้อนเข้าไปประกอบด้วย ความเร็วในการจุ่มลงและดึงขึ้น ระยะทางที่แท่นจับชิ้นงานเคลื่อนที่ เวลาในการเข้าชิ้นงาน เวลาในการพักชิ้นงาน และจำนวนรอบในการจุ่มชิ้นงาน



รูปที่ 2.12 Rotary Encoder [6]

## 2.6 จอแสดงผล GLCD (Graphic Liquid Crystal Display) [7]

เป็นชุด Display ขนาด 128x64 Dot พร้อมบอร์ดควบคุมการแสดงผล โดยใช้คำสั่งแบบ ASCII Command ใช้แหล่งจ่ายไฟเลี้ยงบอร์ด DC 7-12 V มีการทำงาน 3Mode คือ Text Mode, Graphic Mode และ Picture Mode สำหรับโครงงานพิเศษนี้จะใช้เพียง 2 Mode มีดังต่อไปนี้

### 2.6.1 Text Mode

2.6.1.1 สามารถแสดงผลตัวอักษรภาษาอังกฤษ ไม่ต้องเสียเวลาในการสร้าง Font เอง

2.6.1.2 ขนาดของตัวอักษรสำหรับภาษาอังกฤษ 5x7 Dot (กว้างxยาว)ส่วนความกว้างขึ้นอยู่กับตัวอักษรนั้นๆซึ่งทำให้ได้ตัวอักษรที่สวยงาม โดย Font ที่แสดงจะเป็น Font เดียวกับ MS sans Serif ของ Window

2.6.1.3 สามารถกำหนดให้แสดงตัวอักษรในตำแหน่งใดๆบนจอได้ และโดยปกติหนึ่งหน้าจจะแสดงข้อความได้ 4 บรรทัดหรือมากกว่า ส่วนในหนึ่งบรรทัดจะแสดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอักษรได้ก็ตัวนั้น จะขึ้นอยู่กับขนาดความกว้างของตัวอักษรที่ส่งมาในแต่ละตัว ซึ่งจะมีความกว้างไม่เท่ากัน

2.6.1.4 สามารถลบตัวอักษร ในตำแหน่งใดๆบนหน้าจอได้

2.6.1.5 สามารถส่ง Text ข้อความได้ครั้งละ 200 ตัวอักษร(ตัวอักษร+สระ+วรรณยุกต์)

2.6.1.6 สามารถเลือกข้อความได้ว่าจะให้ข้อความที่ส่งมา ชิดซ้าย, ชิดขวาหรืออยู่กึ่งกลาง ของบรรทัดนั้นๆ (ข้อความส่วนที่เกินบรรทัดจะถูกตัดทิ้ง)

2.6.1.7 สามารถเลือกได้ว่าเมื่อข้อความจบบรรทัดแล้วให้ขึ้นบรรทัดใหม่ให้โดยอัตโนมัติ

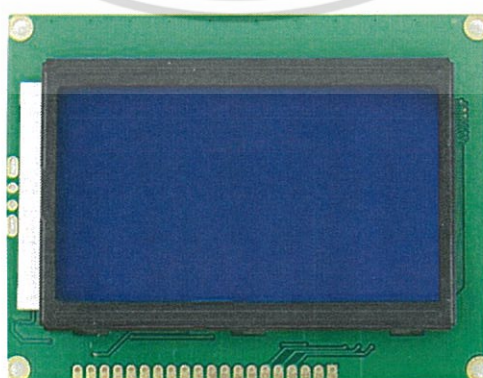
2.6.1.8 เมื่อข้อความเต็มจอข้อความส่วนที่เกินจะถูกตัดทิ้งไป และสามารถส่งเคลียร์หน้าจอ ใหม่ได้

## 2.6.2 Graphic Mode

คำสั่งกลุ่ม Graphic Mode จะเป็นคำสั่งในการวาดรูป สามเหลี่ยมสี่เหลี่ยมรูปสามเหลี่ยม (ไม่เกิน 9 เหลี่ยม) วงรีวงกลมจุดและเส้นตรง ในแนวนอน แนวตั้งและเส้นทแยง ซึ่งรูปเหล่านี้สามารถ วาดให้ปรากฏที่จุดใดๆบนหน้าจอก็ได้ด้วยการกำหนดพิกัดจุด x,y ของรูปให้กับคำสั่งที่ใช้ โดยค่าที่ใช้ กำหนดให้กับพิกัด x จะต้องเป็นค่า ASCII ขนาด 3 Byte ที่เป็น 0-9 (000-127) ส่วนพิกัด y จะต้อง เป็น ASCII ขนาด 2 Byte ที่เป็นตัวเลข 0-9 (00-63) เช่นกัน ซึ่งผู้ใช้สามารถดูค่าตำแหน่งพิกัดของจุด ต่างๆบนหน้าจอได้จากตารางจอภาพในภาคผนวก ซึ่งทำให้สะดวกขึ้นในการกำหนดตำแหน่งพิกัดจุด โดยให้ผู้ใช้มองที่มุมบนซ้ายของหน้าจอ GLCD เป็นพิกัดเริ่มต้น  $x,y = (0,0)$

ลักษณะของรูปที่วาดจะเป็นลักษณะของ ลายเส้น และสามารถ Fill รูปให้เป็นสีดำทั้งรูปได้ ยกเว้นรูปหลายเหลี่ยมจะ Fill เป็นสีดำไม่ได้แต่จะสามารถกำหนดความหนาของเส้นที่วาดได้ สำหรับการ Fill รูปให้เป็นสีดำนั้นจะใช้เวลาในการ Fill พอสมควรโดยเฉพาะรูปสามเหลี่ยมซึ่งอาจจะทำให้ เสียเวลาในการสั่งคำสั่งต่อไปได้

การ Plot รูปออกหน้านั้นจะเป็นในลักษณะการ Plot ไปทับข้อมูลเก่าที่แสดงอยู่บน หน้าจอ จุดใดที่อยู่บนหน้าจอที่ไม่ถูก Plot รูปใหม่ไปทับ จะไม่ถูกลบทิ้ง จะยังคงแสดงให้เห็นอยู่ ซึ่งเป็น ข้อดีที่ผู้ใช้จะสามารถเขียนข้อความลงไปบนรูปที่วาดได้เพียงแต่จะต้องกำหนดพิกัดของรูปหรือข้อความ อย่างไม่ให้ไปทับพิกัดเดิมในตำแหน่งที่เราต้องการจะใช้แสดงผล



รูปที่ 2.13 จอ GLCD 128x64 [5]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

### วิธีการดำเนินงานวิจัย

#### 3.1 การออกแบบระบบฮาร์ดแวร์เครื่อง Dip-Coating

##### 3.1.1 ชุดอุปกรณ์ของโครงงาน



รูปที่ 3.1 ส่วนควบคุมการทำงาน(Arduino UNO R3)

จากรูปที่ 3.1 เป็นส่วนควบคุมการทำงาน ดังได้กล่าวส่วนประกอบต่างๆไว้ในหัวข้อ 2.4.3 ในโครงงานนี้จึงใช้พอร์ตบางส่วนเท่านั้น ได้แก่ 1.ปุ่ม Resetbutton สำหรับเริ่มการทำงานใหม่ 2. พอร์ต USB สำหรับอัปโหลดโปรแกรมเท่านั้น 3.Power jack 12 V สำหรับจ่ายไฟให้กับบอร์ด Arduino UNO R3 4.พอร์ต Power 5V, 3.3Vและ GND 5.พอร์ต Input Analog ใช้พอร์ต A1และ 6. พอร์ต I/O Digital ใช้D2, D3 สำหรับRotary encoder พอร์ต D4, D5, D6, D7 สำหรับจอ GLCD และพอร์ต D10, D11 สำหรับMotor DriverCW และ CCW



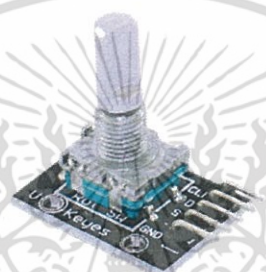
รูปที่ 3.2 ส่วนควบคุมระบบขับเคลื่อน(Motro Driver)

จากรูปที่ 3.2 เป็นส่วนควบคุมระบบขับเคลื่อนมอเตอร์ใช้ไฟกระแสสลับ 110 Vส่งไปยัง Stepping motor 5 Phase โดยใช้โหมดการใช้งานโหมด 2 ป้อนพัลส์ให้กับ CW ต้องการให้หมุนทวนเข็มนาฬิกาหรือตามเข็มนาฬิกาป้อนพัลส์ให้ CCW รายละเอียดจะกล่าวถึงในหัวข้อต่อไป



รูปที่ 3.3 แหล่งจ่ายไฟให้กับ Motor Driver

จากรูปที่ 3.3 เป็นแหล่งจ่ายไฟกระแสสลับให้แก่ Motor Driver 110 V



รูปที่ 3.4 ส่วนควบคุมการสั่งงาน (Rotary Encoder)

จากรูปที่ 3.4 ส่วนควบคุมการสั่งงานติดต่อกับผู้ใช้งานใช้งาน 5 ขา 1. CLK (หมุนตามเข็มนาฬิกา) 2.DT (หมุนทวนเข็มนาฬิกา) 3. SW (สวิตช์) 4. Vcc และ 5.GND สำหรับการต่อจะกล่าวถึงในหัวข้อต่อไป



รูปที่ 3.5 ส่วนการแสดงผล(GLCD)

จากรูปที่ 3.5 เป็นส่วนแสดงผลขนาด 128x64 Dot ใช้แหล่งจ่ายไฟเลี้ยงบอร์ด AC/DC 7-12 V มีการทำงาน 3 โหมด โคร่งงานนี้ใช้แค่ 2 โหมดคือ Text Mode และ Graphic Mode รายละเอียดได้กล่าวไว้ในหัวข้อ 2.6 โดยใช้ขา 1, 2, 3, 4, 5, 6, 15, 17, 19 และ 20 หลักการเชื่อมต่อ

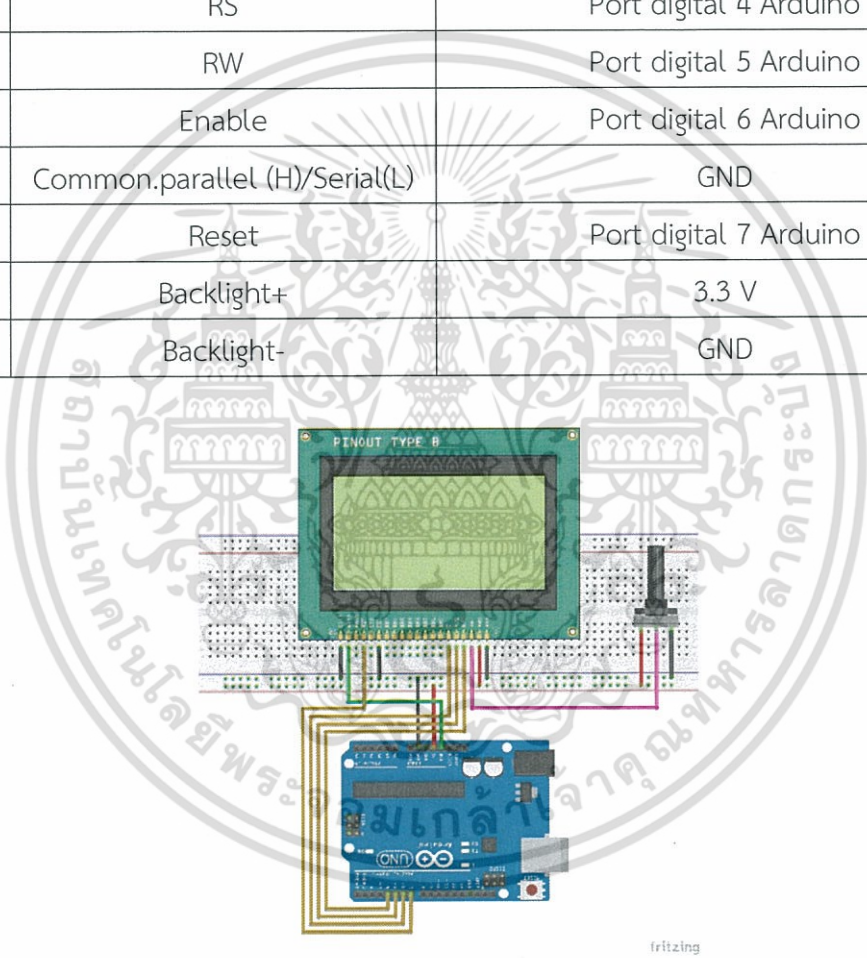
จะกล่าวในหัวข้อต่อไป สำหรับการสั่งงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2 หลักการเชื่อมต่ออุปกรณ์

#### 3.2.1 แสดงการต่อจอ GLCD

ตารางที่ 3.1 แสดงฟังก์ชันของจอ GLCD

Pin	Function	Connect
1	GND	GND
2	Vcc	5 Volts
3	Ajuste de contrate	Pin Potentiometer
4	RS	Port digital 4 Arduino
5	RW	Port digital 5 Arduino
6	Enable	Port digital 6 Arduino
15	Common.parallel (H)/Serial(L)	GND
17	Reset	Port digital 7 Arduino
19	Backlight+	3.3 V
20	Backlight-	GND



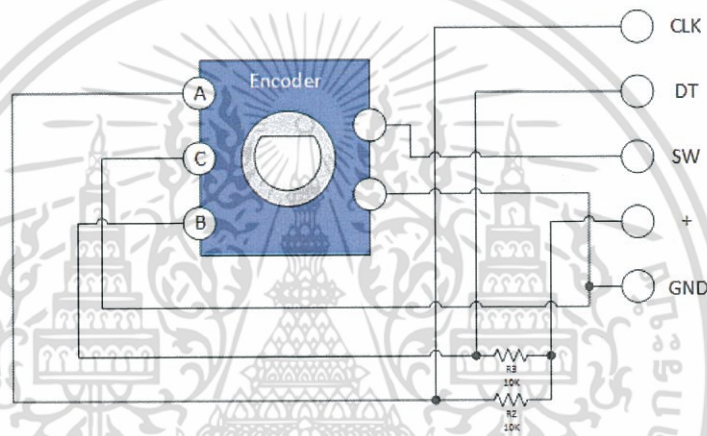
รูปที่ 3.6 การเชื่อมต่อจอGLCD กับ Arduino

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

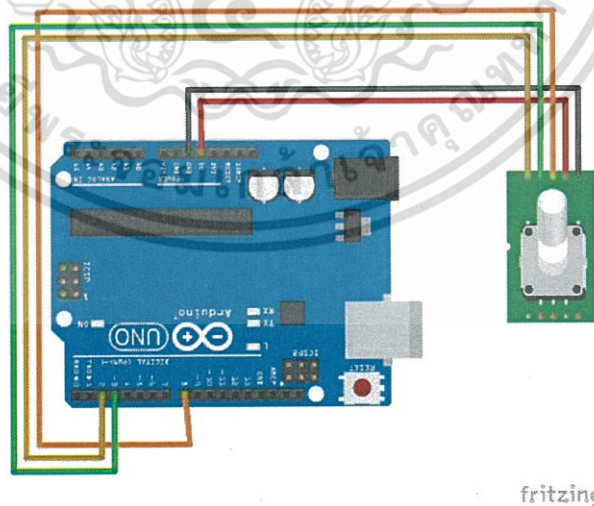
### 3.2.2 แสดงการต่อ Rotary Encoder

ตารางที่ 3.2 แสดงฟังก์ชันการใช้งานของ Rotary Encoder

Rotary Encoder	Function	Connect
A	CLK	Port digital 2 Arduino
B	DT	Port digital 3 Arduino
C	SW	Port Analog A1 Arduino
D	Vcc	Port digital 8 Arduino
E	GND	GND



รูปที่ 3.7 ไดอะแกรม Rotary Encoder [9]

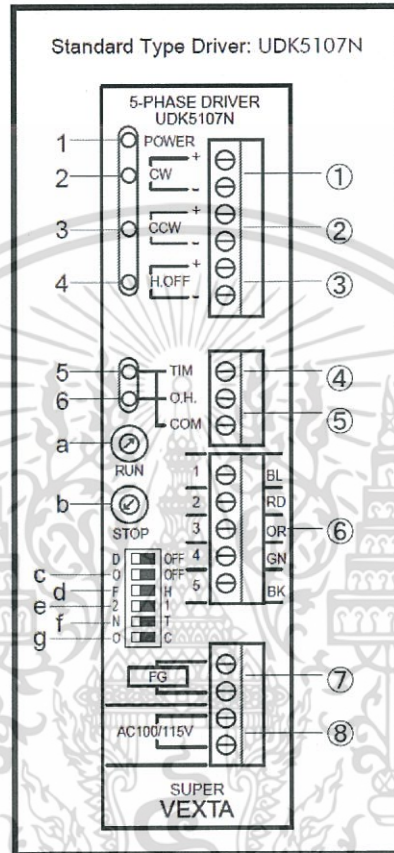


รูปที่ 3.8 การเชื่อมต่อ Rotary Encoder กับ Arduino

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3 ระบบขับเคลื่อน (Driver) [10]

สเตปปีงมอเตอร์จะทำงานโดยผ่านชุดขับเคลื่อน(Motor Driver) เพื่อสร้างสัญญาณขับเคลื่อนมอเตอร์โดยสร้างเป็นสัญญาณพัลส์ ให้ทำงานตามต้องการ ซึ่งความละเอียดของการขับอยู่ที่ 0.72 องศาคือ 500 พัลส์/สเตป อธิบายการทำงานแต่ละโหมดที่ใช้ในการขับเคลื่อนมอเตอร์

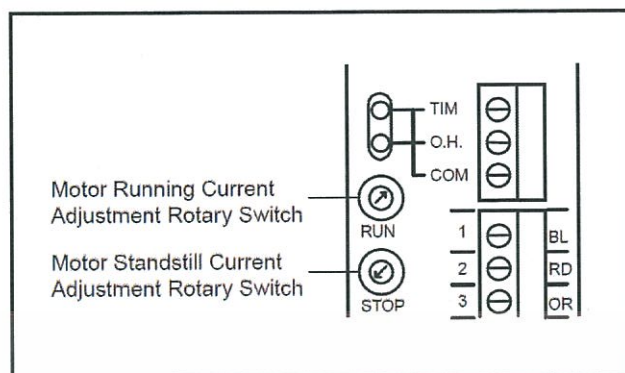


รูปที่ 3.9 ชุดขับเคลื่อนสเตปปีงมอเตอร์ (Motor Driver)

จากรูปที่ 3.9 โหมด ① CW Pulse Signal Input Terminal คือป้อนพัลส์ให้กับ CW ถ้าต้องการให้มอเตอร์หมุนทวนเข็มนาฬิกาและตามเข็มนาฬิกาจะต้องจ่าย High และ Low ให้กับ CCW โหมด ② CCW Pulse Signal Input Terminal คือ ถ้าอยากให้มอเตอร์หมุนตามเข็มนาฬิกาจ่ายพัลส์ให้กับ CW ถ้าอยากให้มอเตอร์หมุนทวนเข็มนาฬิกาจ่ายพัลส์ให้กับ CCW โหมด ⑥ Motor Connection Terminal คือเป็นการเชื่อมต่อระหว่าง Diver กับ Motor และโหมด ⑧ Power Source Connection Terminal คือเป็นแหล่งจ่ายไฟฟ้าให้กับ Motor Driver รุ่น UDK5107N ต้องการไฟฟ้ากระแสสลับที่ 100-115 V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.1 Motor Running Current Adjustment Rotary Switch



รูปที่ 3.10 โหมดการใช้งานในการจ่ายกระแสให้กับ (Motor Driver)

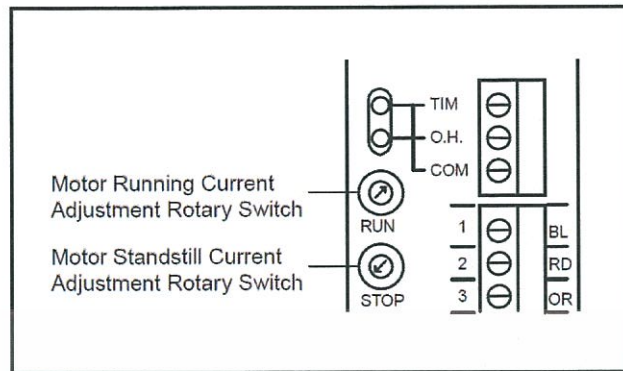
RUN Switch Settings	Running Current [A/phase]		
	UDK5107N	UDK5114N UDK5114N-M	UDK5128N UDK5128N-M
0	0.21	0.45	0.76
1	0.25	0.52	0.90
2	0.28	0.58	1.04
3	0.32	0.64	1.17
4	0.36	0.71	1.31
5	0.39	0.77	1.44
6	0.43	0.83	1.58
7	0.46	0.89	1.72
8	0.50	0.96	1.85
9	0.54	1.02	1.99
A	0.57	1.09	2.13
B	0.61	1.15	2.26
C	0.64	1.21	2.39
D	0.68	1.27	2.53
E	0.72	1.33	2.67
F	0.75	1.40	2.80

รูปที่ 3.11 การจ่ายกระแสให้กับ Motor Driver

จากรูปที่ 3.11 เป็นสวิตช์แบบหมุน ตั้งค่าไว้ที่ F เนื่องจากสเปคมอเตอร์ที่นำมาใช้งานขับกระแสอยู่ที่ 0.75 A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.2 Motor Standstill Current Adjustment Rotary Switch



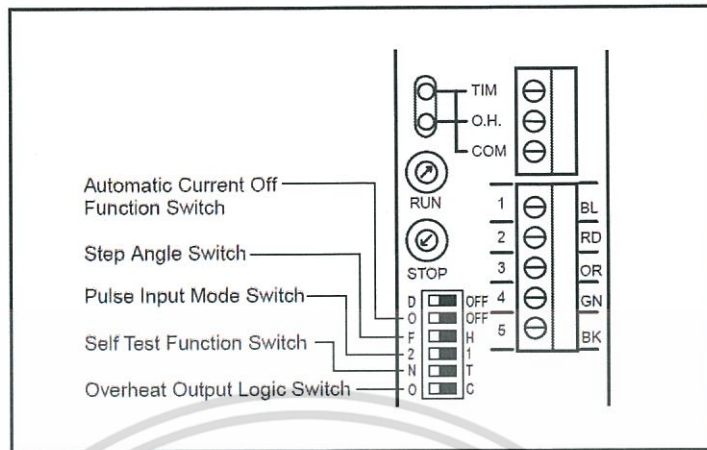
รูปที่ 3.12 โหมดการใช้งานในการจ่ายกระแสให้กับ (Motor Driver)

STOP Switch Settings	% of Running Current		
	UDK5107N	UDK5114N UDK5114N-M	UDK5128N UDK5128N-M
0	10	19	9
1	10	19	9
2	10	19	9
3	15	21	14
4	23	28	21
5	29	34	28
6	37	40	36
7	43	47	43
8	51	54	50
9	57	61	57
A	65	67	64
B	72	74	71
C	79	80	79
D	86	87	86
E	93	94	93
F	100	100	100

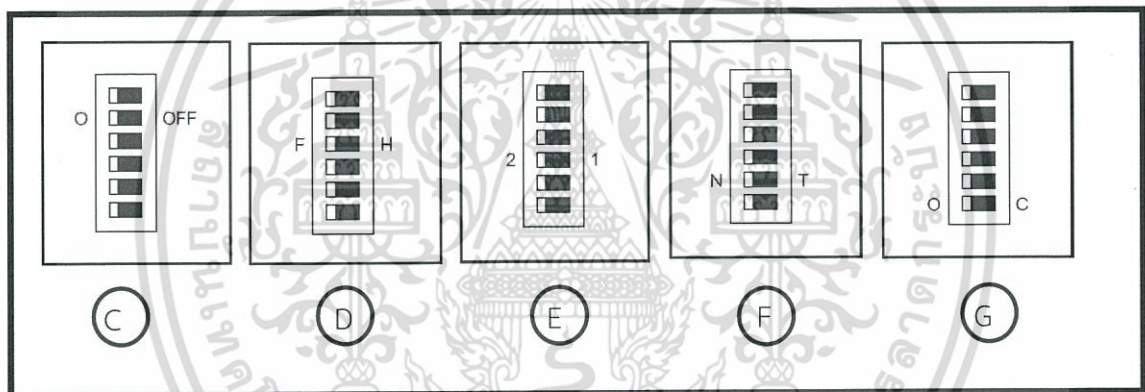
รูปที่ 3.13 เปอร์เซนต์การจ่ายกระแสให้กับ UDK5107N

จากรูปที่ 3.13 เป็นสวิตช์แบบหมุนจ่ายกระแสให้มอเตอร์ขณะหยุดนิ่ง แม้ว่ามอเตอร์จะหยุด แต่ก็มีการจ่ายกระแสไฟฟ้าให้อยู่ ดังนั้น % การหยุดนิ่งของกระแสขั้นต่ำให้อยู่ที่ 40% เพื่อให้เป็นมาตรฐานจึงตั้งค่าที่ 7

3.3.3 การปรับฟังก์ชันการใช้งาน



รูปที่ 3.14 ฟังก์ชันการใช้งานในการทดสอบ



รูปที่ 3.15 ฟังก์ชันการใช้งานในการทดสอบแต่ละโหมด

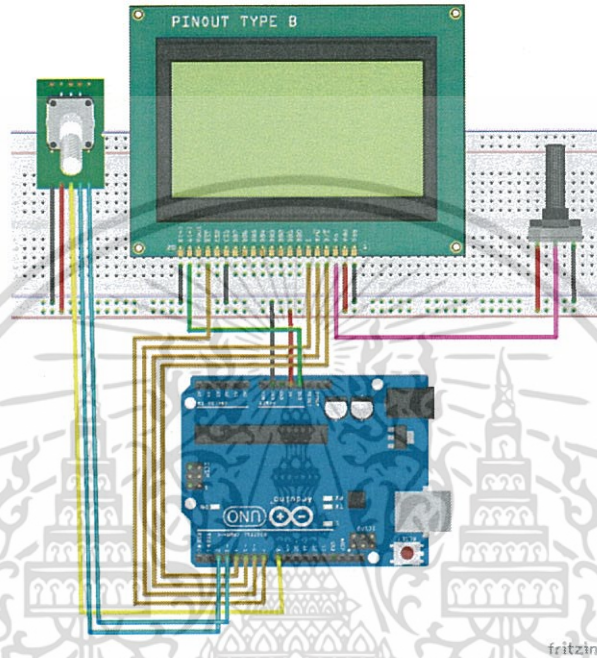
จากรูปที่ 3.15 โหมด C ปรับไปที่ [O] เนื่องจากการป้องกันการเสียหายจากการขับมอเตอร์ ถ้าอุณหภูมิสูงกว่า 80 องศาเซลเซียส สัญญาณจะถูกตัด และในส่วนของ [OFF] ถ้าอุณหภูมิภายในเกิน 80 องศาเซลเซียส มอเตอร์จะทำงานต่อไปเรื่อยๆโดยไม่คำนึงถึงสัญญาณโหมด D ปรับไปที่ [F] คือ Full 1 Step = 0.72 องศา (1 รอบ=500 พัลส์) เนื่องจากการปรับให้ตรงกับมอเตอร์ที่ใช้ และในส่วนของ [H] Half 1 Step = 0.36 องศา (1 รอบ=1000พัลส์) โหมด E [2] CCW ถ้าอยากให้มอเตอร์หมุนตามเข็มนาฬิกาจ่ายพัลส์ให้กับ CW ถ้าอยากให้มอเตอร์หมุนทวนเข็มนาฬิกาจ่ายพัลส์ให้กับ CCW และ[1]CW คือป้อนพัลส์ให้กับ CW ถ้าต้องการให้มอเตอร์หมุนทวนเข็มนาฬิกาและตามเข็มนาฬิกาจะต้องจ่าย High และ Low ให้กับ CCW โหมด F ปรับไปที่ [N] Norm เป็นการใช้งานจริง และ[T] Test เป็นการทดสอบการเชื่อมต่อสัญญาณโหมด G ปรับไปที่ [O] เมื่อมีอุณหภูมิเพิ่มขึ้น 80 องศา มอเตอร์จะหยุดขับและปรับอุณหภูมิให้อยู่ที่อุณหภูมิห้องแล้วทำงานต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4 การออกแบบระบบซอฟต์แวร์เครื่อง Dip-Coating

#### 3.4.1 การออกแบบส่วนติดต่อกับผู้ใช้

ประกอบด้วยอุปกรณ์น้อยชิ้น เพียงแค่จอ GLCD, Rotary Encoder ทั้งสองจะทำการติดต่อกับผู้ใช้โดยมีหน้าจอเป็นสิ่งที่แสดงการทำงานของเครื่อง หน้าจอที่ใช้ดังกล่าวมีลำดับการทำงานดังรูปที่ 3.16

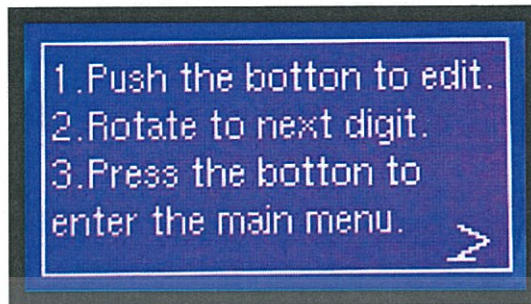


รูปที่ 3.16 การเชื่อมต่อ จอ GLCD, Rotary Encoder กับ Arduino

จากรูปที่ 3.16 เมื่อเราใช้ Rotary Encoder ในการสั่งงานจากผู้ใช้เช่น เมื่อต้องการใส่ค่าพารามิเตอร์ เพื่อให้ระบบฮาร์ดแวร์ทำงานตามความต้องการของผู้ใช้ จากนั้นค่าที่เราใส่แสดงบนหน้าจอของ GLCD ส่วนค่า R 10K ปรับค่าได้จะเป็นส่วนของการปรับความสว่างของหน้าจอ เมื่อความสว่างเป็นไปตามต้องการของผู้ใช้แล้วสามารถนำ R 10k ปรับค่าได้ออกจากวงจรได้

### 3.4.2 การทำงานของระบบซอฟต์แวร์

#### 3.4.2.1 การแสดงผลทางจอ GLCD ของระบบสังเคราะห์ฟิล์มแบบจุ่มเคลือบ



(a)



(b)

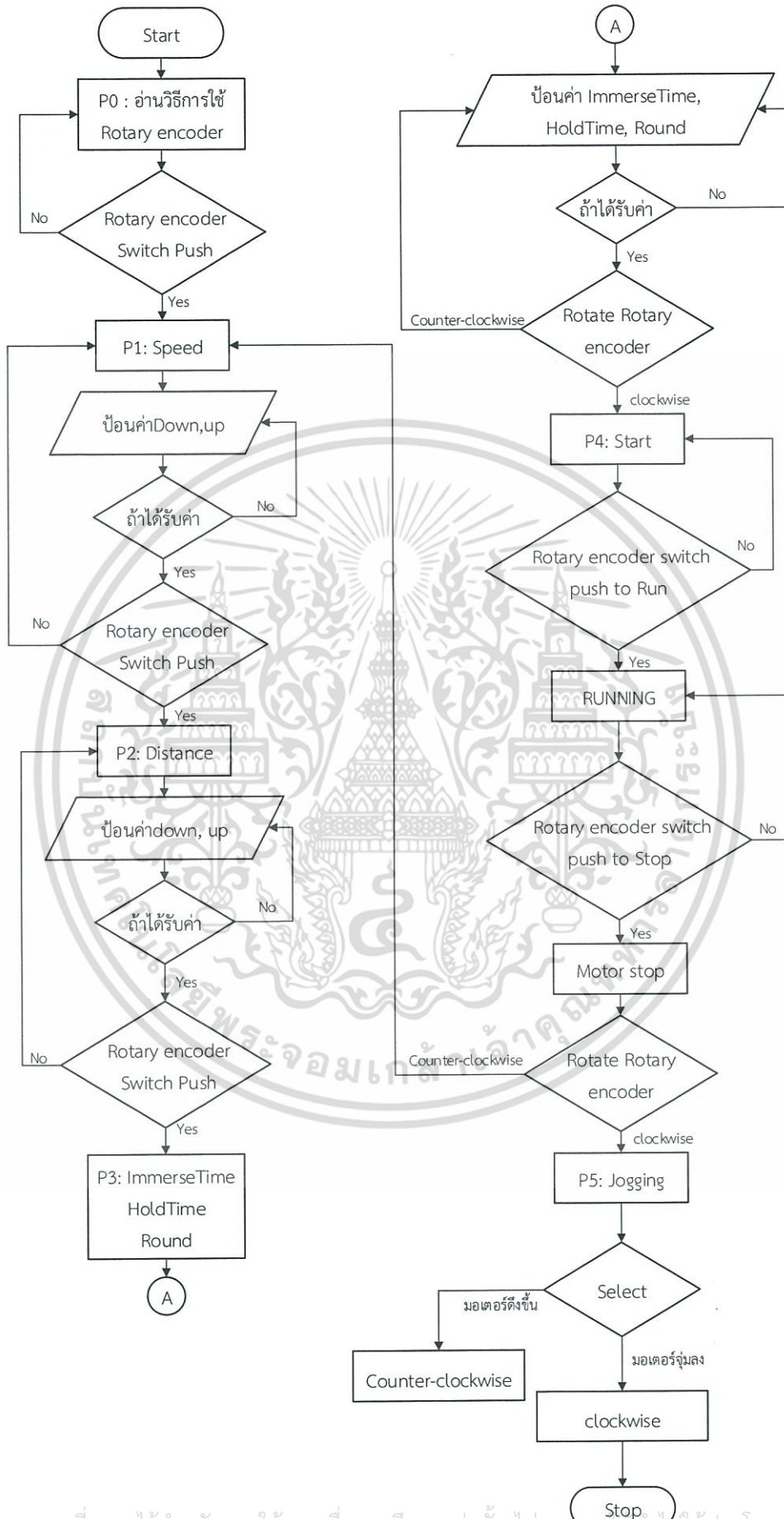
รูปที่ 3.17 ตัวอย่างแสดงผลทางจอ GLCD ของระบบสังเคราะห์ฟิล์มแบบจุ่มเคลือบ

จากรูปที่ 3.17 เป็นตัวอย่างการแสดงผลทางจอ GLCD ของระบบสังเคราะห์ฟิล์มบางโดยใช้ Rotary encoder ในการสั่งงาน หน้าแรกคือP0:เป็นการแสดงวิธีการตั้งค่าข้อมูลต่างๆ เมื่อกดสวิตช์ 1 ครั้งจะเข้าสู่หน้าP1:Dip-coat โดยการตั้งค่าความเร็วในการจุ่มลงและดึงขึ้นของแผ่นสไลด์ หน้าP2:Distance ตั้งค่าระยะทางในการเคลื่อนที่ของมอเตอร์ หน้าP3:Immerse time(เวลาแช่สไลด์), Holding time(เวลาพักสไลด์)และ Round (รอบในการจุ่ม) เมื่อตั้งค่าต่างๆเรียบร้อยแล้วเข้าสู่ หน้า P4:Start กด 1 ครั้งระบบจุ่มเคลือบจะทำงานเมื่อต้องการหยุด กดสวิตช์ 1 ครั้ง และหน้าP5:Jogging เมื่อต้องการให้ชิ้นงานอยู่ตำแหน่งที่เราต้องการ โดยกดค้างที่ตำแหน่งนั้นเมื่อต้องการหยุดก็ปล่อยสวิตช์ เป็นต้น

#### 3.4.2.2 การใช้งานส่วนควบคุมการสั่งงาน

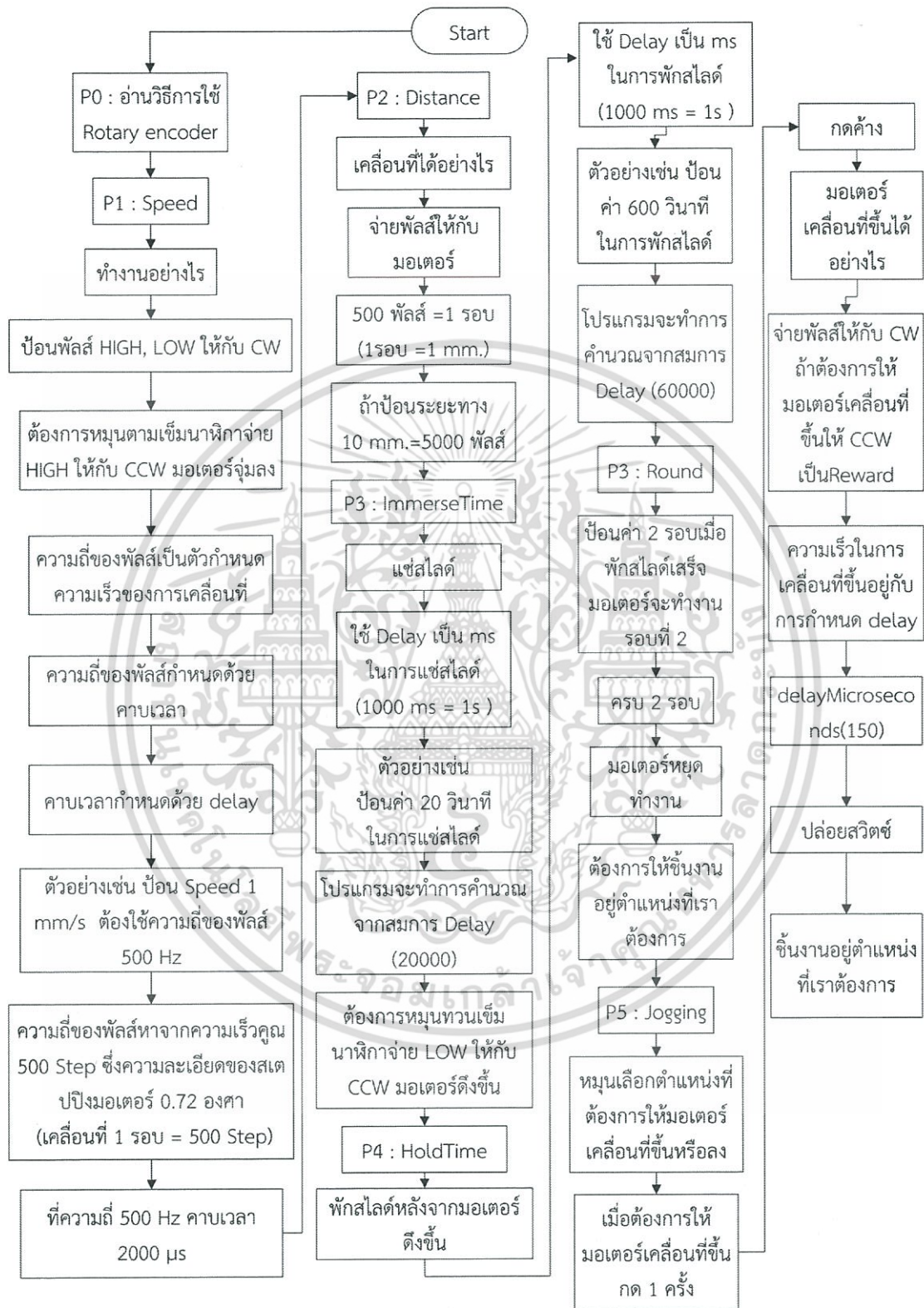
การสั่งงานของระบบขึ้นอยู่กับ Rotary encoder สั่งผ่านจอ GLCD เพื่อให้เข้าใจการทำงาน ดังได้อธิบายดังรูป 3.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้า  
 รูปที่ 3. 18 ผังงาน(Flowchart)ส่วนควบคุมการสั่งงาน  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ต่อสาธารณะโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2.3 ไตอะแกรมการสร้างระบบสังเคราะห์ฟิล์มแบบจุ่มเคลือบ



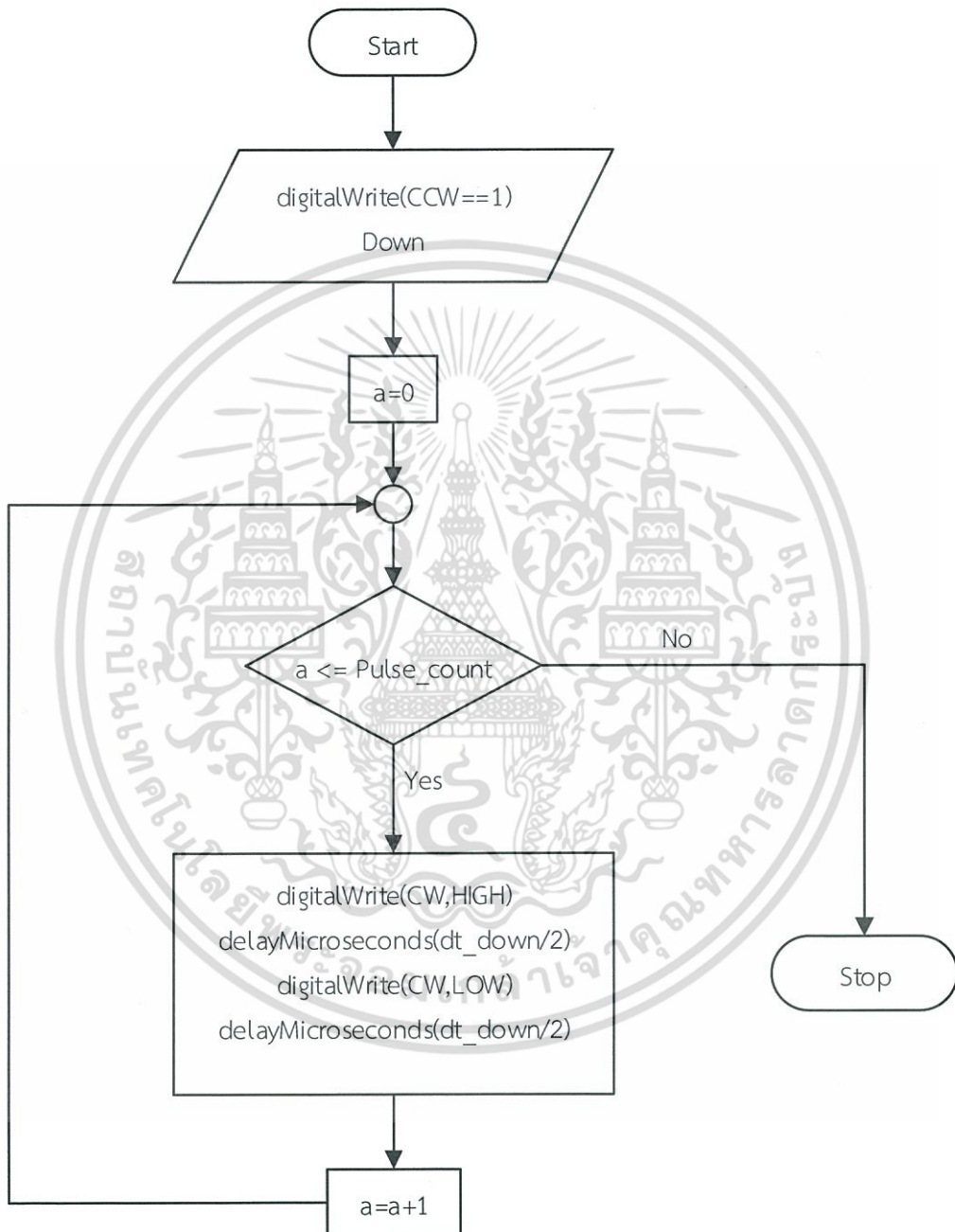
รูปที่ 3.19 ไตอะแกรมการทำงานระบบสังเคราะห์ฟิล์มแบบจุ่มเคลือบ

จากรูปที่ 3.19 เป็นการทำงานของระบบสังเคราะห์ฟิล์มแบบจุ่มเคลือบ เพื่อให้ผู้ใช้เข้าใจถึงเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาก็เท่านั้น เมื่อนำไปใช้เผยแพร่ในเชิงพาณิชย์โดยไม่ผ่านการอนุญาตของเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4.2.4 กระบวนการทำงานของสเตปปีงมอเตอร์(Stepping motor)

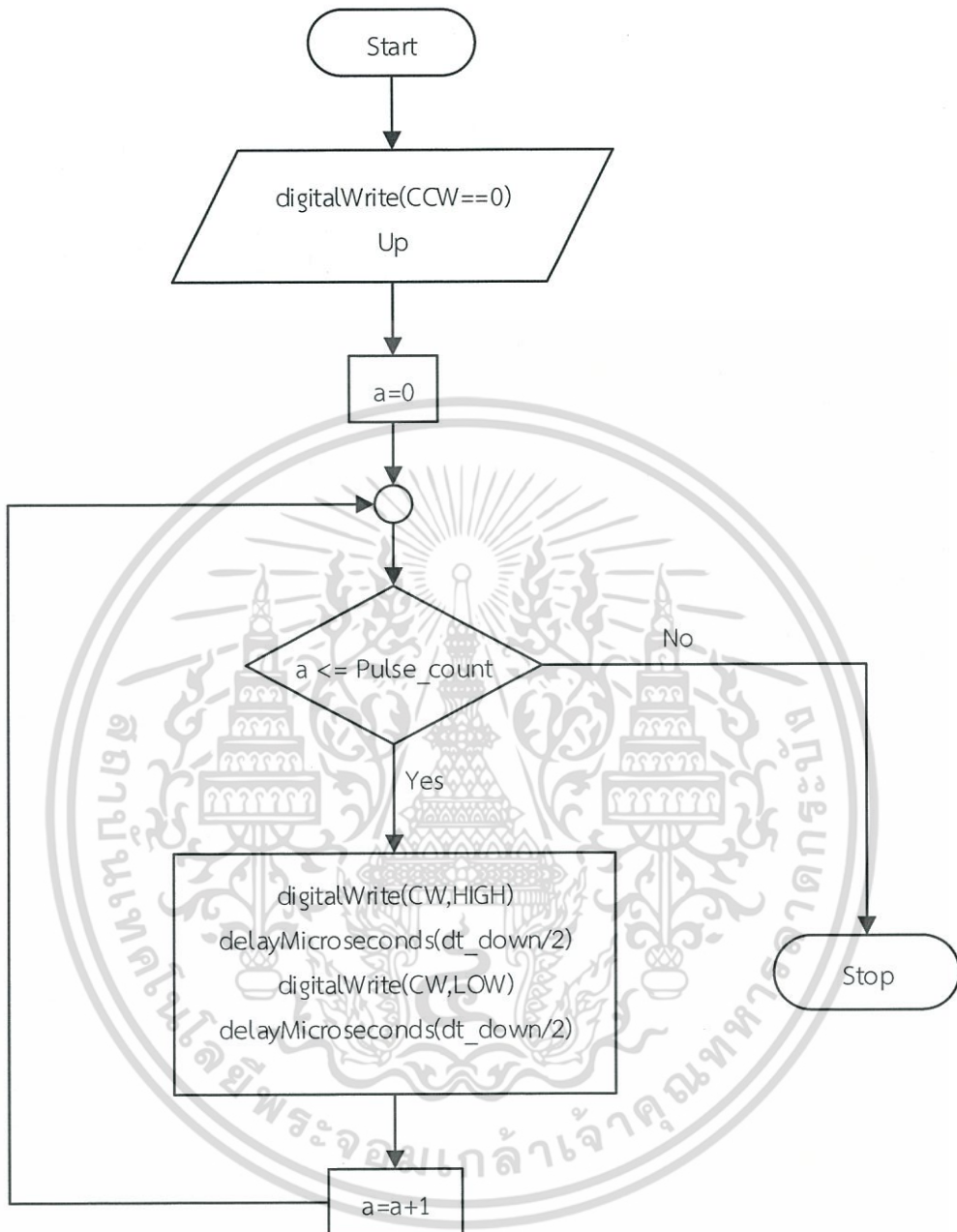
สเตปปีงมอเตอร์จะทำงานได้โดยการจ่ายพัลส์ให้กับ Motor Driver ดังนั้นในส่วน  
ของซอฟต์แวร์การเคลื่อนที่ของมอเตอร์จะอธิบายดังต่อไปนี้

#### 3.4.2.4.1 การเคลื่อนที่ของมอเตอร์ขณะจุ่มลง



รูปที่ 3.20 การเคลื่อนที่ของมอเตอร์ขณะจุ่มลง

## 3.4.2.4.2 การเคลื่อนที่ของมอเตอร์ขณะตั้งขึ้น

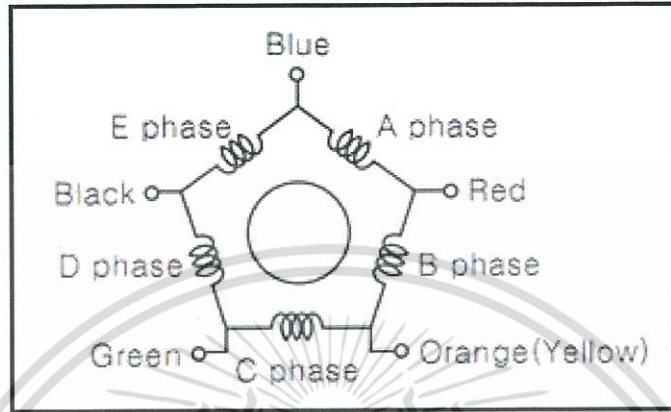


รูปที่ 3.21 การเคลื่อนที่ของมอเตอร์ขณะตั้งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.5 โครงสร้าง Stepping Motor 5 Phase

สเตปมอเตอร์ชนิด 5 เฟสที่ใช้ในโครงงานนี้ สามารถหมุนไปได้ 0.72 องศาต่อ 1 สเตปและใช้กระแสในการขับแต่ละขด 0.75 แอมแปร์ มีโครงสร้างดังรูปที่ 3.18 วิธีการขับจะเป็นแบบไปโพลาร์ โดยการขับแต่ละรอบประกอบด้วย 10 สเตป วิธีการขับแต่ละสเตปแสดงในตารางที่



รูปที่ 3.22 โครงสร้าง 5 Phase Stepping Motor [11]

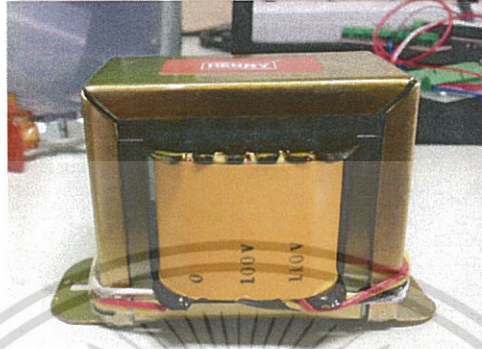
ตารางที่ 3.3 แสดงวิธีการขับ Stepping Motor ชนิด 5 เฟส

step	0	1	2	3	4	5	6	7	8	9
VOHGA	H	H	L	L	L	L	L	L	L	H
VOHGB	L	H	H	H	L	L	L	L	L	L
VOHGC	L	L	L	H	H	H	L	L	L	L
VOHGD	L	L	L	L	L	H	H	H	L	L
VOHGE	L	L	L	L	L	L	L	H	H	H
VOLA	L	L	L	L	H	H	H	L	L	L
VOLB	L	L	L	L	L	L	H	H	H	L
VOLC	H	L	L	L	L	L	L	L	H	H
VOLD	H	H	H	L	L	L	L	L	L	L
VOLE	L	L	H	H	H	L	L	L	L	L
CW	→									
CCW	←									

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.6 แหล่งจ่ายไฟกระแสตรง

ในการขับเคลื่อนมอเตอร์ใช้การจ่ายไฟฟ้า 110V และ 12V เพื่อใช้ขับเคลื่อนมอเตอร์(Driver)และไมโครคอนโทรลเลอร์ Arduino UNO R3 ดังนั้นใช้หม้อแปลงในการแปลงไฟฟ้าจาก 220V เป็น 110V ให้กับ DC Power Supply



รูปที่ 3.23 แหล่งจ่ายไฟฟ้ากระแสสลับ 110v

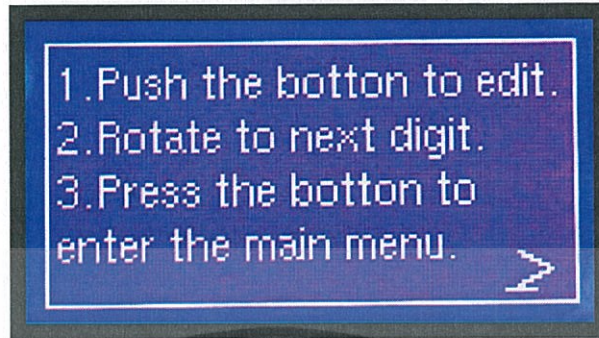


รูปที่ 3.24 แหล่งจ่ายไฟฟ้ากระแสตรง 12-16 V

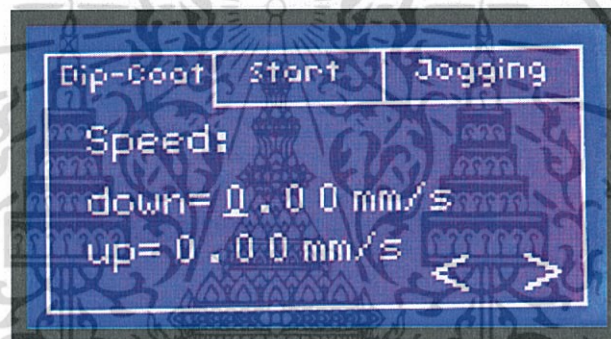
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.7 การแสดงผล (GLCD)

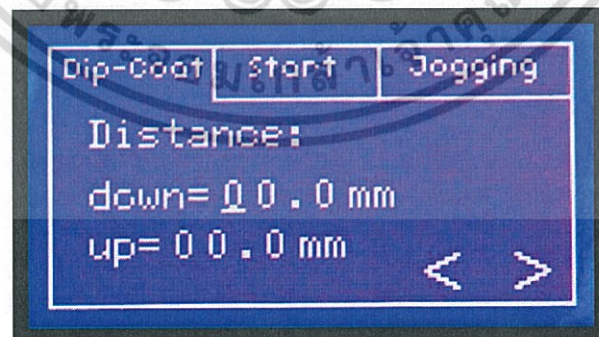
#### 3.7.1 ผลการทำงานของซอฟต์แวร์ส่วนการติดต่อกับผู้ใช้ ในส่วนของการแสดงผลออกทางจอ GLCD 128x64



รูปที่ 1 หน้าแรกอ่านวิธีการใช้ โดยการกด Rotary encoder เพื่อเข้าสู่การตั้งค่าข้อมูล

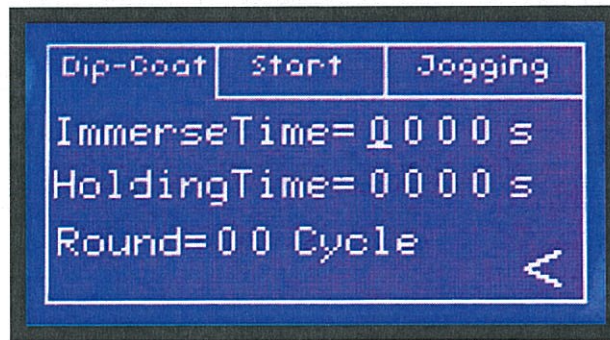


รูปที่ 2 ตั้งค่าความเร็วในการจุ่มและความเร็วในการดึงขึ้นตามต้องการ เมื่อใส่ค่าเรียบร้อยแล้ว กด ">" เพื่อไปหน้า Distance และกด "<" เพื่อกลับไปหน้าแรก



รูปที่ 3 ตั้งค่าระยะทางที่เราต้องการให้มอเตอร์เคลื่อนที่หรือระยะที่เราต้องการจุ่ม มอเตอร์สามารถเคลื่อนที่ได้ระยะทางสูงสุด 9 เซนติเมตร กด ">" เพื่อไปหน้าถัดไป กด "<" เพื่อไปหน้า Speed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4 ตั้งค่าเวลาในการแช่แผ่นรองรับ เวลาในการพักเพื่อให้แผ่นรองรับแห้งและจำนวนรอบในการจุ่ม ถ้าต้องการความหนามากขึ้นสามารถกำหนดให้จุ่มหลายๆรอบกด “<” เพื่อไปหน้า Distance

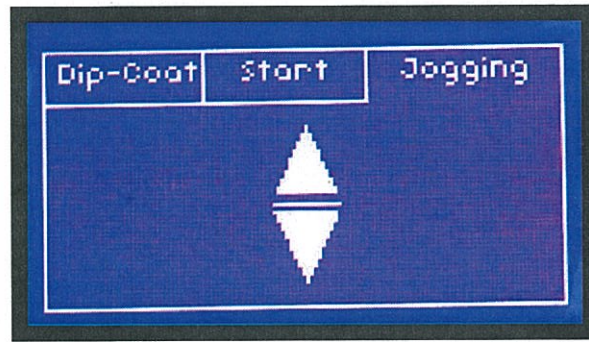


รูปที่ 5 กด 1 ครั้ง เส้นใต้จะปรากฏที่ start เมื่อกดอีก 1 ครั้งมอเตอร์จะเคลื่อนที่ตามที่เราระบุไว้



รูปที่ 6 เมื่อกด start จะขึ้น Running เพื่อแสดงถึงการเคลื่อนที่ของมอเตอร์ ในขณะที่มอเตอร์กำลังทำงานสามารถกด Rotary 1 ครั้งเพื่อหยุดการทำงานของมอเตอร์ขณะนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7 ถ้าต้องการให้มอเตอร์เคลื่อนที่ขึ้นและลงตามที่เรากำหนด

กด ▲ ค้างมอเตอร์จะเคลื่อนที่ขึ้น

กด ▼ ค้างมอเตอร์จะเคลื่อนที่ลง

รูปที่ 3.25 แสดงผลของจอ GLCD 128 x 64

### 3.8 วัสดุเครื่องมือ

#### 3.8.1 แผ่นรองรับ

แผ่นรองรับที่ใช้ในการทดลองเป็นแผ่นกระจกสไลด์ (Microscope glass slide) ขนาดกว้าง 2.5 cm. x 7.5 cm. และมีความหนา 1 mm. เป็นกระจกที่ผลิตในประเทศเยอรมัน ซึ่งมีลักษณะที่ใสและผิวเรียบเหมาะสำหรับนำมาทำวิจัย โดยแผ่นรองรับที่ใช้ในการทดลองต้องมีการทำความสะอาดพื้นผิวก่อน สามารถเตรียมได้ดังขั้นตอนต่อไปนี้

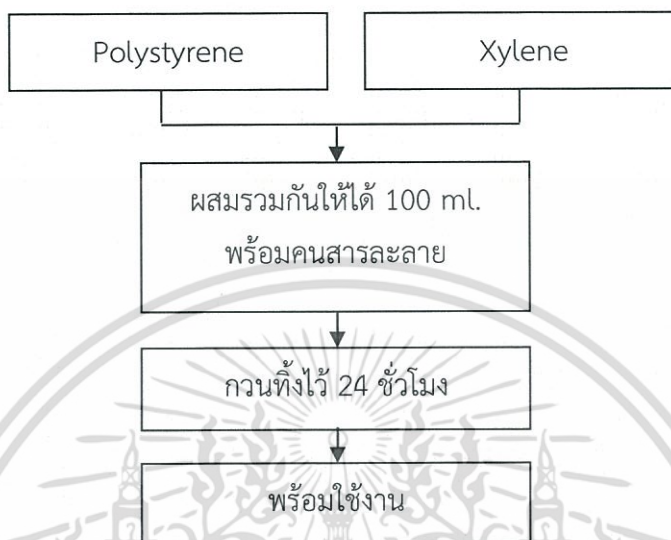
1. ล้างทำความสะอาดฟิล์ม แล้วนำไปเป่าด้วยแก๊สไนโตรเจน
2. ล้างด้วยอะซิโตนเป็นเวลา 20 นาที
3. ล้างด้วยเมทานอลเป็นเวลา 20 นาที
4. ล้างด้วยน้ำ DI เป็นเวลา 20 นาที
5. นำไปเป่าให้แห้งด้วยแก๊สไนโตรเจน
6. นำไปอบที่อุณหภูมิ 120 องศาเซลเซียส เป็นเวลา 30 นาที

เหตุผลที่ต้องการมีการล้างพื้นผิวของวัสดุก่อนนำไปเคลือบ มีดังนี้

1. เพื่อกำจัดหรือป้องกันการมีชั้นปนเปื้อนของสารอื่นบนผิวหน้าของวัสดุที่ทำให้เกิดการยึดติดกันน้อยลง
2. เพื่อให้โมเลกุลของหยดของเหลวกับผิวหน้าของวัสดุสัมผัสกันได้มากที่สุด
3. เพื่อเพิ่มแรงตึงผิวบนพื้นผิวของวัสดุซึ่งจะทำให้ผลการเปียกผิวดีขึ้น (Good wetting)

### 3.8.2 สารละลาย

ในการทดลอง สารเคมีที่ใช้ในการผสมสารละลายประกอบไปด้วยสารเคมีหลัก 2 อย่างคือ Polystyrene และ Xylene สารละลายที่ได้จะเป็นสารละลายพอลิเมอร์ โดยในการเตรียมสารละลาย สำหรับใช้ในการทดลอง เตรียมได้ดังต่อไปนี้ ดังรูป

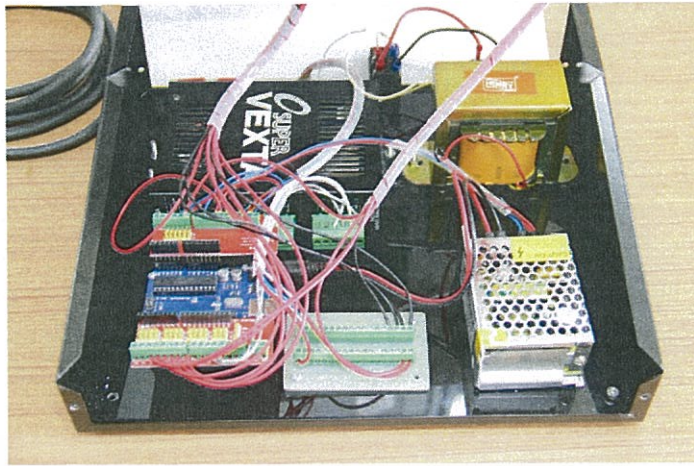


รูปที่ 3.26 ขั้นตอนการเตรียมสารละลาย Polystyrene

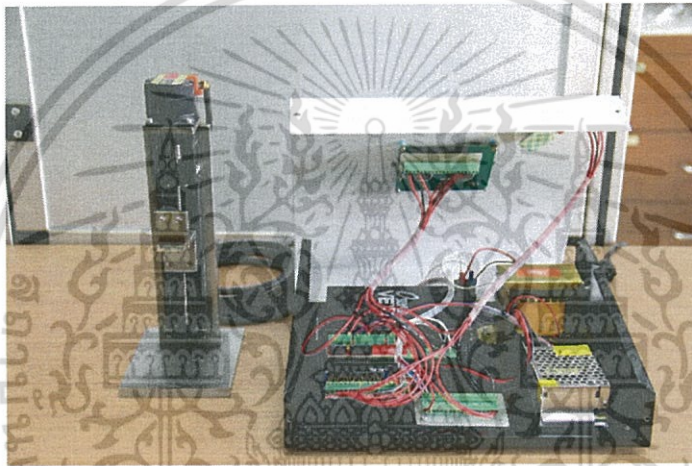
### 3.9 ระบบเคลือบฟิล์มแบบจุ่มเคลือบ

ระบบเคลือบฟิล์มแบบจุ่มที่ใช้ในการทดลองนั้นเป็นเครื่องที่ได้ออกแบบและสร้างขึ้นมา โดยการทำงานของระบบนั้นเป็นการควบคุมสเตปมอเตอร์ด้วยไมโครคอนโทรลเลอร์ Arduino UNO R3 ผ่านชุดขับมอเตอร์ Driver และส่งงานผ่าน Rotary encoder พร้อมแสดงผลผ่านจอ GLCD ส่วนประกอบภายในระบบมีดังนี้

1. แหล่งจ่ายไฟฟ้ากระแสสลับ 110 V (Transformer)
2. แหล่งจ่ายไฟฟ้ากระแสตรง 12 V (DC Power supply)
3. ชุดขับเคลื่อน (Driver)
4. ไมโครคอนโทรลเลอร์ (Arduino UNO R3)



(a)



(b)

รูปที่ 3.27 โครงสร้างภายในของระบบเคลือบแบบจุ่ม



รูปที่ 3.28 ระบบเคลือบฟิล์มแบบจุ่มเคลือบ

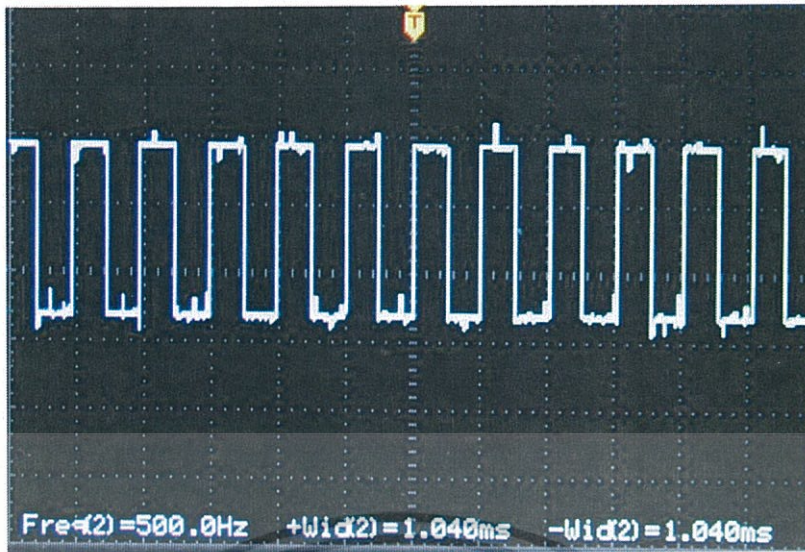
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.10 ความเร็วในการจุ่มเคลือบ

การเคลือบแบบจุ่ม (Dip Coating) เป็นวิธีการหนึ่งของการประยุกต์ใช้ในการเตรียมฟิล์มบาง โดยอาศัยหลักการเคลือบสารละลายบนพื้นผิวอย่างช้าๆ การจุ่มจะนำสารละลายมาเคลือบบนพื้นผิวอย่างช้าๆ ซึ่งต้องมีการควบคุมความเร็วในการดึงให้คงที่ เพื่อให้ผิวของวัสดุมีความเรียบสม่ำเสมอและความหนาของฟิล์มน้อย หากต้องการฟิล์มที่มีความหนาน้อยให้ใช้ความเร็วในการเคลือบแบบจุ่มเร็ว แต่ถ้าต้องการเคลือบแบบหนามากให้ทำการจุ่มหลายๆ ครั้งเพื่อเพิ่มความหนาให้กับฟิล์มได้ ดังนั้นจึงหาความเร็วในการจุ่มที่เหมาะสมดังตารางที่ 3.4

ตารางที่ 3.4 ความสัมพันธ์ของความเร็วกับความถี่

Speed (mm/s)	Frequency (Hz) จากการคำนวณ	Frequency (Hz) จากการทดลอง	+Width	-Width
0.1	50	49.02	10.00ms	10.00ms
0.5	250	250	2.000ms	2.040ms
1.0	500	500	1.040ms	1.040ms
1.5	750	735.3	680.0us	680.0us
2.0	1000	976.6	512.0us	512.0us
2.5	1250	1220	410.0us	410.0us
3.0	1500	1471	340.0us	340.0us
3.5	1750	1724	300.0us	280.0us
4.0	2000	1923	250us	250us
4.5	2250	2155	232.0us	232.0us
5.0	2500	2404	208.0us	208.0us
5.5	2750	2604	192.0us	192.0us
6.0	3000	2841	176.0us	176.0us



รูปที่ 3.29 ตัวอย่างการวัดค่าจากออสซิลโลสโคปที่ความเร็ว Speed 1 mm/s

### 3.11 การทดสอบระบบการจุ่มเคลือบ

จากที่ได้สร้างระบบสังเคราะห์ฟิล์มแบบจุ่มเคลือบขึ้น เพื่อเป็นไปตามความต้องการของผู้ใช้จึงทดสอบระบบ โดยทดสอบความเร็วที่แตกต่างกัน ระยะทางเท่ากัน ดังนั้นถ้าความเร็วเพิ่มขึ้นระยะทางเท่าเดิม เวลาในการเคลื่อนที่ก็จะน้อยลง ดังตารางที่ 3.5

ตารางที่ 3.5 ความสัมพันธ์ระหว่างความเร็วกับเวลาในการเคลื่อนที่

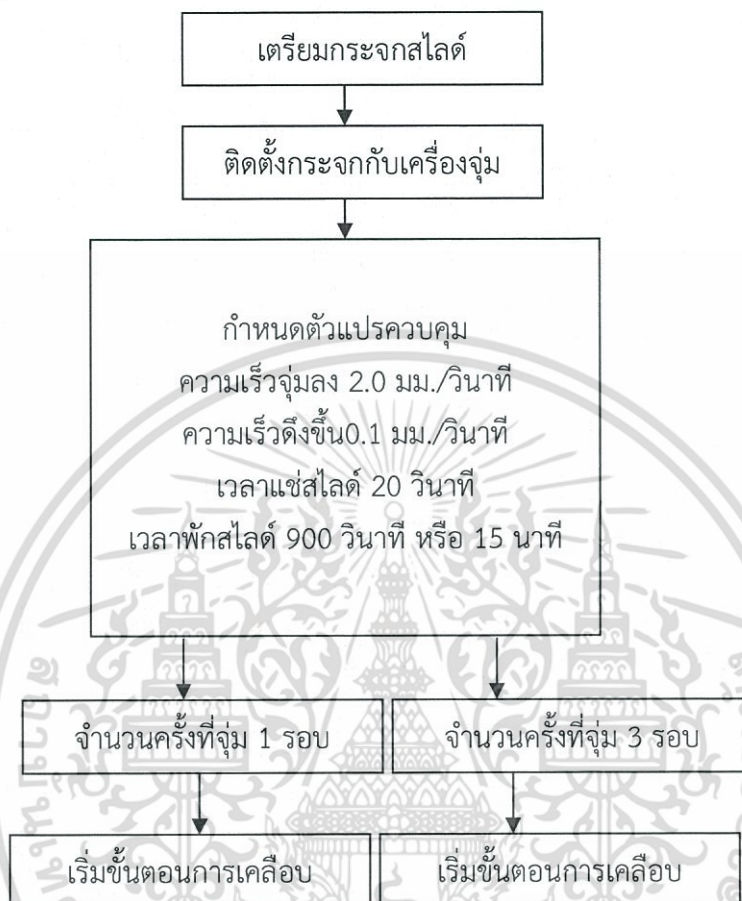
Speed (mm/s)	Distance (mm)	Time(s) จำนวน	Time (s) ทดลอง			Time(s) เฉลี่ย	Speed(mm/s) จากการวัด
			1	2	3		
0.1	50	500	508	506	506	506.67	0.09
0.5	50	100	104	103	104	103.67	0.48
1.0	50	50	52.7	53.7	53.7	53.37	0.94

จากตารางที่ 3.5 ที่ความเร็ว 0.1 มม./วินาที เวลาในการเคลื่อนที่เฉลี่ย 506.67 วินาที เมื่อเพิ่มความเร็ว 0.5 มม./วินาที เวลาในการเคลื่อนที่เฉลี่ยลดลง 103.67 วินาที และเมื่อเพิ่มความเร็ว 1.0 มม./วินาทีเวลาในการเคลื่อนที่เฉลี่ยลดลงอีก 53.37 วินาที

### 3.12 ขั้นตอนกระบวนการเคลือบ

กระบวนการเคลือบฟิล์มพอลิเมอร์บนกระจกสไลด์ ได้แบ่งการทดลองออกเป็น 3 การทดลอง

#### การทดลองที่ 1



รูปที่ 3.30 แผนภาพขั้นตอนกระบวนการเคลือบฟิล์มในการทดลองที่ 1

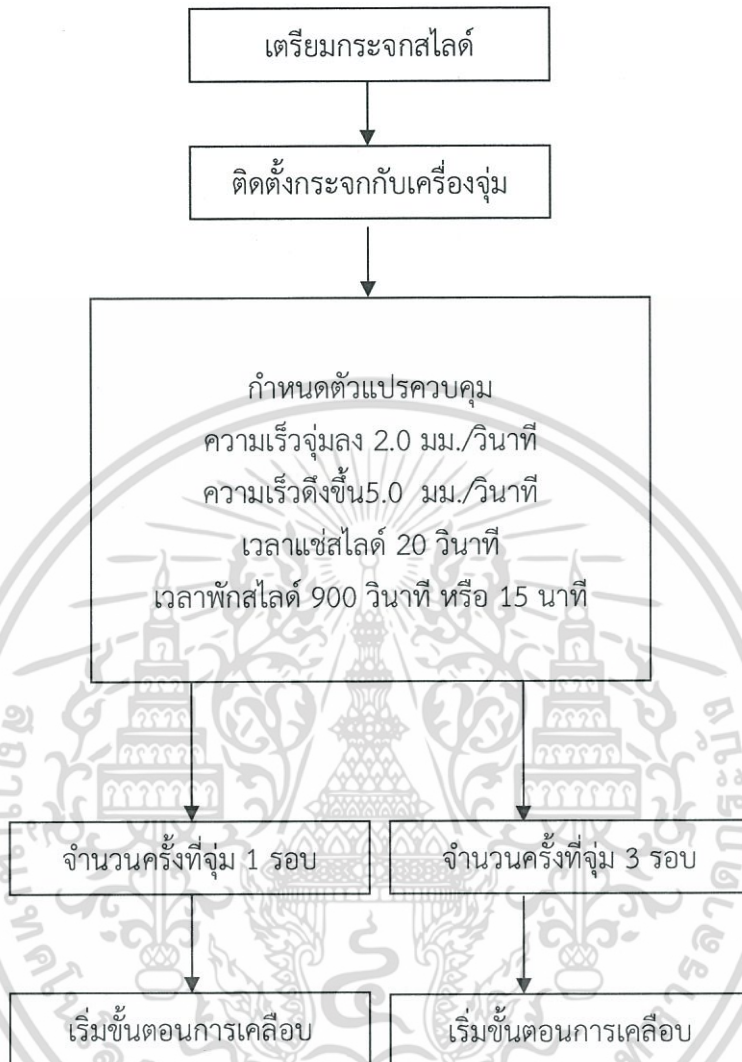
## การทดลองที่ 2



รูปที่ 3.31 แผนภาพขั้นตอนกระบวนการเคลือบฟิล์มในการทดลองที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การทดลองที่ 3



รูปที่ 3.32 แผนภาพขั้นตอนกระบวนการเคลือบฟิล์มในการทดลองที่ 3

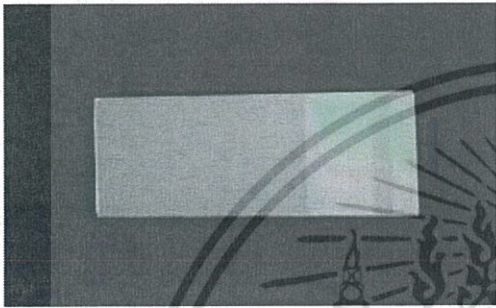
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4 ผลการวิจัยและการอภิปรายผล

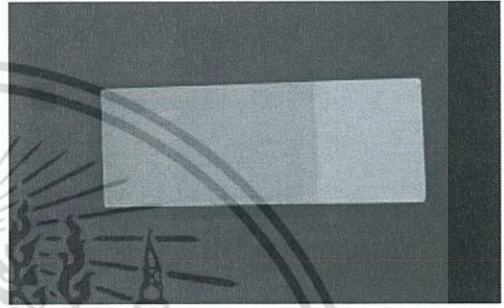
### 4.1 ผลการทดลอง

#### 4.1.1 ผลการทดลองการเคลือบฟิล์มแบบจุ่มเคลือบ

การทดลองที่ 1 ความเร็วที่ใช้ในการดึงขึ้น 0.1 มม./วินาที  
ความเร็วที่ใช้ในการจุ่มลง 2.0 มม./วินาที

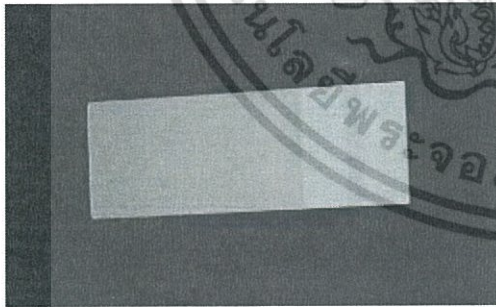


จำนวนครั้งในการจุ่ม 1 รอบ

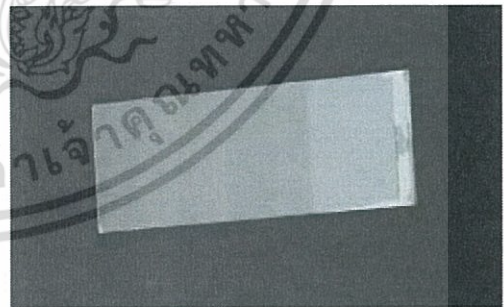


จำนวนครั้งในการจุ่ม 3 รอบ

การทดลองที่ 2 ความเร็วที่ใช้ในการดึงขึ้น 2.0 มม./วินาที  
ความเร็วที่ใช้ในการจุ่มลง 2.0 มม./วินาที



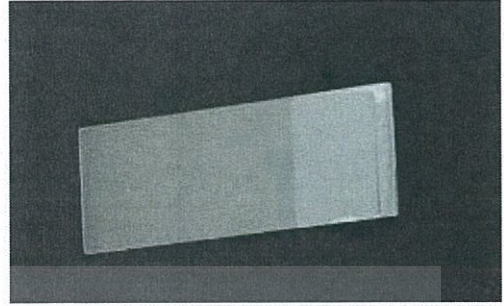
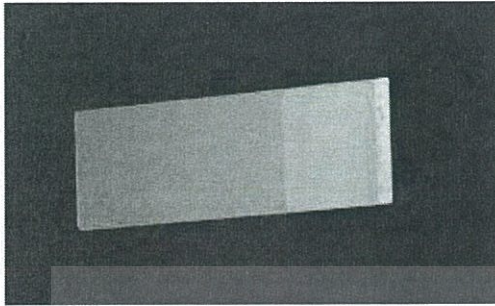
จำนวนครั้งในการจุ่ม 1 รอบ



จำนวนครั้งในการจุ่ม 3 รอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 3 ความเร็วที่ใช้ในการดึงขึ้น 5.0 มิลลิเมตร/วินาที  
 ความเร็วที่ใช้ในการจุ่มลง 2.0 มิลลิเมตร/วินาที



จำนวนครั้งในการจุ่ม 1 รอบ

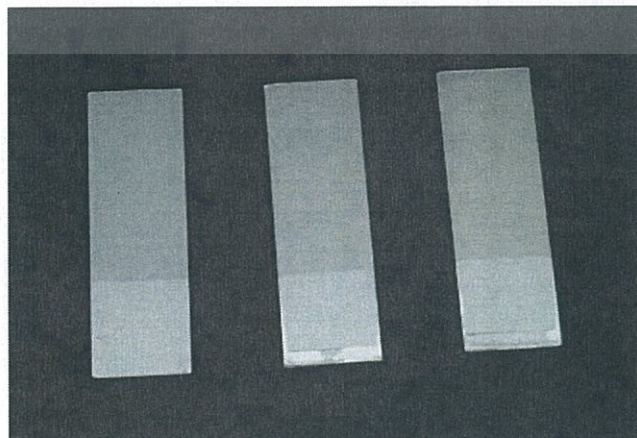
จำนวนครั้งในการจุ่ม 3 รอบ

#### 4.1.2 การเปรียบเทียบในการจุ่มเคลือบแต่ละรอบ

4.1.2.1 การจุ่มเคลือบฟิล์มจำนวน 1 รอบ ความเร็วในการดึงขึ้นแตกต่างกัน



4.1.2.2 การจุ่มเคลือบฟิล์มจำนวน 3 รอบ ความเร็วในการดึงขึ้นแตกต่างกัน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.1.3 การอภิปรายผล

จากการทดลองใช้ความเร็วในการดึงขึ้นที่แตกต่างกันพบว่าความเร็วในการดึงขึ้น 0.1 มม./วินาที จะทำให้ฟิล์มเป็นริ้ว เมื่อเพิ่มความเร็วขึ้น 2.0 มม./วินาที ฟิล์มที่เคลือบกระจกจะบางลง พื้นผิวเรียบขึ้นแต่ตรงปลายกระจกจะเห็นเป็นฟิล์มหนาๆ เนื่องจากเกิดการย้อยของสารละลาย และเมื่อเพิ่มความเร็วขึ้นเป็น 5.0 มม./วินาที ฟิล์มที่เคลือบกระจกจะบางลงอีก พื้นผิวก็เรียบขึ้นด้วยเนื่องจากมีการดึงกระจกขึ้นเร็ว แต่ก็ทำให้สารละลายไปรวมกันที่จุดหยุดไหลเยอะขึ้น ทั้งการย้อยของสารละลายนี้ขึ้นอยู่กับสารละลายที่เรานำมาใช้งาน จากการทดลองจะเห็นได้ว่ารอบในการจุ่มฟิล์มมีผลต่อความหนาของฟิล์ม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

# สรุปผลการวิจัยและข้อเสนอแนะ

### 5.1 สรุปผลการวิจัย

ในการศึกษาทดลองเพื่อพัฒนาการออกแบบซอฟต์แวร์และฮาร์ดแวร์เครื่อง Dip-Coating เพื่อทำให้ง่ายต่อการใช้งานมากขึ้น โดยสิ่งที่พัฒนาขึ้นมาคือ การใช้จอแสดงผล GLCD การควบคุมการสั่งงานด้วย Rotary Encoder และใช้คอนโทรลเลอร์ Arduino UNO R3 ในการควบคุมโปการสั่งทั้งหมด ส่วนที่ใช้ในการขับเคลื่อนคือ สเตปปีงมอเตอร์ชนิด 5 เฟส มีองศาการหมุน 0.72 องศาต่อสเตป สเตปปีงมอเตอร์ที่ในการทดลองนี้สามารถเลื่อนไปได้ 90 มิลลิเมตร ใช้กระแสไฟฟ้าในขดลวด 0.75 แอมแปร์ และเขียนโปรแกรมไมโครคอนโทรลเลอร์ เพื่อสร้างชุดคำสั่งต่างๆขึ้นมาซึ่งจะสั่งงานหรือแก้ไขค่าต่างๆผ่านทาง Rotary Encoder พร้อมทั้งแสดงผลออกทางหน้าจอ GLCD เมื่อทดสอบโปรแกรมโดยใช้ Rotary Encoder ในการควบคุมการทำงานสามารถทำได้โดยใช้คำสั่ง Tick มาแทนการใช้ Delay เพื่อการประมวลผลได้ดีและทำให้โปรแกรมมีความเสถียรมากขึ้น

การพัฒนาซอฟต์แวร์กับโปรแกรม Arduino

1. การใช้ Tick แทน Delay ใน loop ทำให้เกิดผลลัพธ์ที่ดีคือ โปรแกรมไม่หยุดประมวลผลไปชั่วขณะ แต่ถ้าหากใช้ Delay โปรแกรมจะหยุดการประมวลผลชั่วขณะ นั่นคือสิ่งที่เราต้องการ
2. การใช้ฟังก์ชัน Tick เป็นการทำงานแบบมัลติทาสกิ้ง(Multitasking) ซึ่งคือการทำงานหลายๆงานไปพร้อมกัน และเป็นอิสระต่อกัน โดยจะเป็นการแบ่งงานแต่ละงานด้วยการเขียนโปรแกรมเป็นฟังก์ชันโดยเป็นฟังก์ชันที่ได้สร้างขึ้นมาเองเพื่อตอบสนองต่อการกดสวิตช์ของ Rotary encoder
3. ฟังก์ชัน Tick คือ ช่วงเวลาในการเข้าไปทำงานในฟังก์ชันที่กำหนด มีหน่วยเป็นมิลลิวินาที ถ้าไม่มีการระบุพารามิเตอร์นี้ จะกำหนดค่าเป็น 5 มิลลิวินาที
4. การพัฒนาซอฟต์แวร์และฮาร์ดแวร์ ของบอร์ด Arduino ที่มีการพัฒนาแบบ Open Source ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษาจนถึงผู้ที่พัฒนาโปรแกรมขั้นสูง และใช้งานได้ฟรีมีประโยชน์ต่อการพัฒนาสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ด หรือโปรแกรมได้อีกด้วย

เมื่อได้ทำการทดลองเคลือบฟิล์มสามารถสรุปผลได้ดังนี้

1. ถ้าต้องการฟิล์มที่บางๆ สามารถกำหนดความเร็วในการดึงกระจกสไลด์ให้เร็วขึ้น จากการทดลองความเร็ว 5 มม/วินาที ทำให้ฟิล์มบางขึ้นทั้งนี้ขึ้นกับความหนืดของสารละลายที่นำมาทดลอง
2. ถ้าต้องการเคลือบฟิล์มที่มีความหนาหลายๆชั้น สามารถสั่งงานให้จุ่มได้หลายๆรอบตามต้องการ
3. จากการจุ่มแต่ละครั้งจะมีรอยต่อระหว่างชั้นฟิล์ม ซึ่งเป็นตัวแปรที่ทำให้เกิดผลต่างๆตามมา การปรับสภาพด้วยความร้อนหลังการเคลือบด้วยวิธีการจุ่มจึงยังคงต้องมีการศึกษาต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## เอกสารอ้างอิง

- [1] Inventor. 2013. **สตีปเปอร์มอเตอร์**. [Online].  
Available:<http://www.inventor.in.th/home/สตีปเปอร์มอเตอร์/#.WNEllLV97Df>
- [2] VT-sun. 2012. **กระบวนการเคลือบฟิล์มบาง (Thin film process)**. [Online].  
Available:[http://www.vtsun.com/index.php?option=com\\_content&view=article&id=10&Itemid=9&lang=th](http://www.vtsun.com/index.php?option=com_content&view=article&id=10&Itemid=9&lang=th)
- [3] บริษัท วินัส ซัพพลาย จำกัด. 2012. **Arduino คืออะไร**. [Online].  
Available:<http://www.thaieasyelec.com/article-wiki/basic-electronics/บทความ-arduino-คืออะไร-เริ่มต้นใช้งาน-arduino.html>
- [4] Arduitrronic. 2017. **Rotary Encoder Module Brick Sensor**. [Online].  
Available: <https://www.arduitronics.com/product/147/rotary-encoder-module-brick-sensor>
- [5] ArduinoAll. 2017. **LCD 128x64 Dots Graphic Blue Color**. [Online].  
Available: <https://www.arduinoall.com/product/68/lcd-128x64-dots-graphic-blue-color-backlight-lcd-display-shield-5-0v>
- [6] www.arduino.in.th Rotary. 2017. **Encoder Module**. [Online].  
Available:<https://www.arduino.in.th/product/259/rotary-encoder-module>
- [7] บริษัท อีทีที จำกัด. 2017. **คู่มือ ET-REMOTE GLCD V1**. [Online].  
Available: file:///C:/Users/Arm2017/Downloads/Documents/man-ET-REMOTE-GLCD-V1.pdf
- [8] นางสาวรัชดาพร ประทุมวัลย์งานวิจัยฉบับสมบูรณ์เรื่อง การศึกษาโครงสร้างและสมบัติของฟิล์มบางและผงสังกะสีออกไซด์ที่เติมด้วย โคบอลต์โดยเทคนิคโซล-เจล ปีพ.ศ.2555, คณะวิทยาศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี. [Online].  
Available: file:///C:/Users/Amcomputer/Downloads/Fulltext%232\_271129.pdf
- [9] RedWaves theme by Themient2015. **Keyes Rotary Encoder Schematic**. [Online].  
Available:<http://henrysbench.capnfatz.com/henrys-bench/arduino-sensors-and-input/keyes-ky-040-arduino-rotary-encoder-user-manual/>
- [10] ORIENTAL MOTOR CO.LTD. 2000. **5 Phase Stepping Motor and Drive**. [Online].  
Available:file:///C:/Users/Arm2017/Downloads/VEXTA.pdf
- [11] Becerro.2011. **โครงสร้าง Stepping motor**. [Online].  
Available:<http://ba-k.net/foro/threads/ayuda-con-motor-paso-a-paso-de-5-fases.1910791/>
- [12] YTC America Inc. 2008. **Schematic diagram of the Dip-Coating Process**  
Available: [http://www.ytca.com/dip\\_coating](http://www.ytca.com/dip_coating)
- [13] H. Schmidt, M. Mennig.2000. **Wet Coating Technologies for Glass**. [Online].  
Available: <http://www.solgel.com/articles/nov00/mennig.htm>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ก

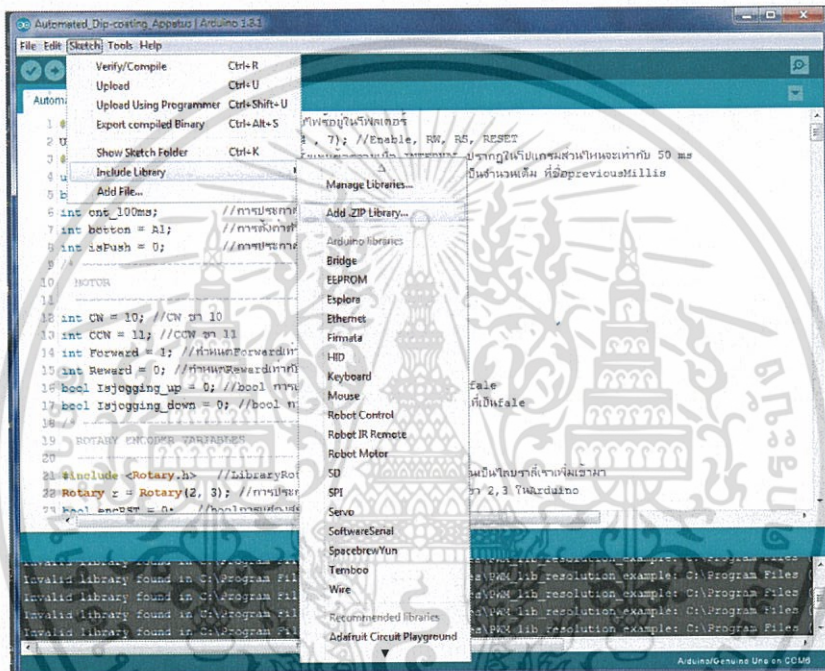
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ก

### วิธีการลงไลบรารี (Library) ให้กับโปรแกรม Arduino IDE

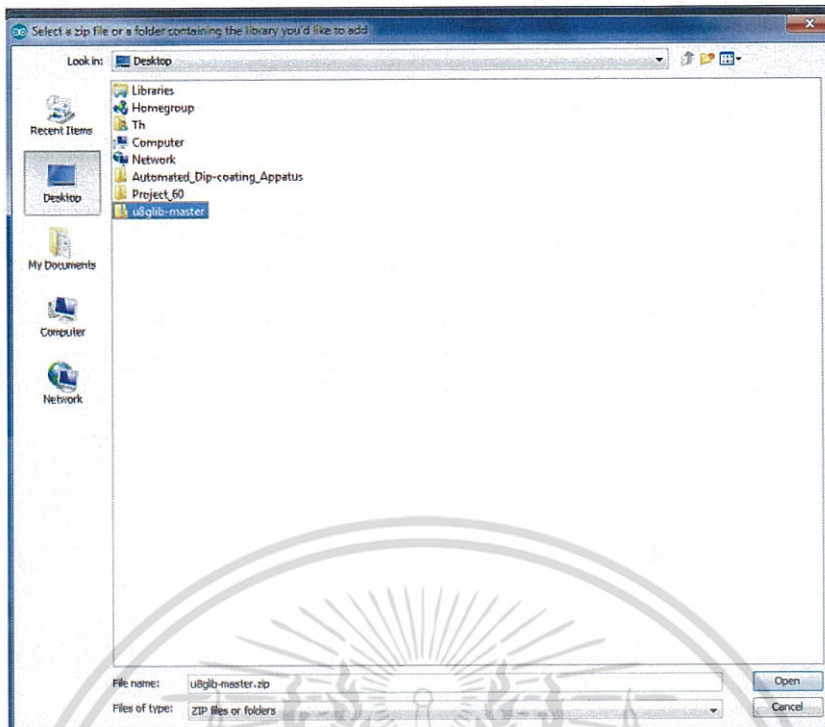
Library คือ โปรแกรมที่ทำให้ sensor และโมดูลต่างๆ ใช้งานได้ง่ายขึ้นสะดวกต่อการใช้งาน และไม่มีความซับซ้อนมากนัก ซึ่งการใช้งาน Library บางตัวผู้ใช้งานจะต้องเพิ่มเข้าไปด้วยตนเอง ดังวิธีการต่อไปนี้

#### 1.วิธีการลงไลบรารี (Library) สำหรับ จอ GLCD จะใช้ไลบรารี U8glib

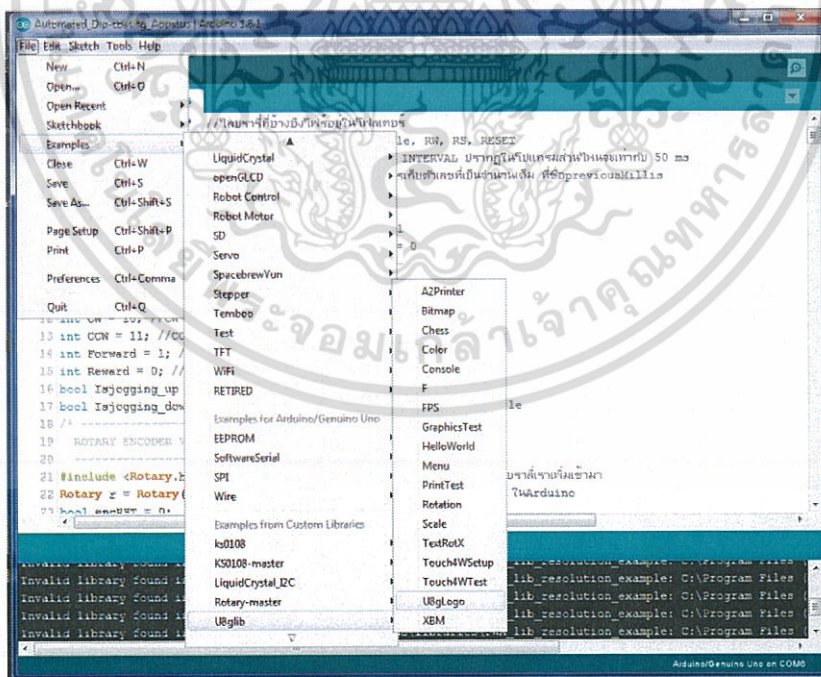


รูปที่ 1 เปิดโปรแกรม Arduino IDE ขึ้นมา แล้วกดไปที่ Sketch > Include Library กดเลือก Include Library > Add ZIP Library

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

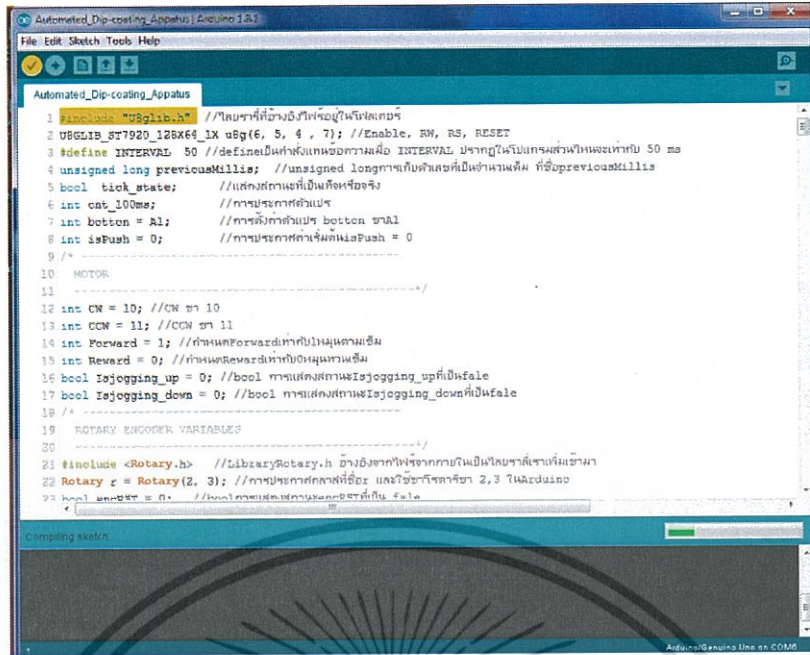


รูปที่ 2 เปิด Folder หรือตำแหน่งที่เก็บ Library เอาไว้ โดยกดเลือกเฉพาะ Folder ของ Library กด Open

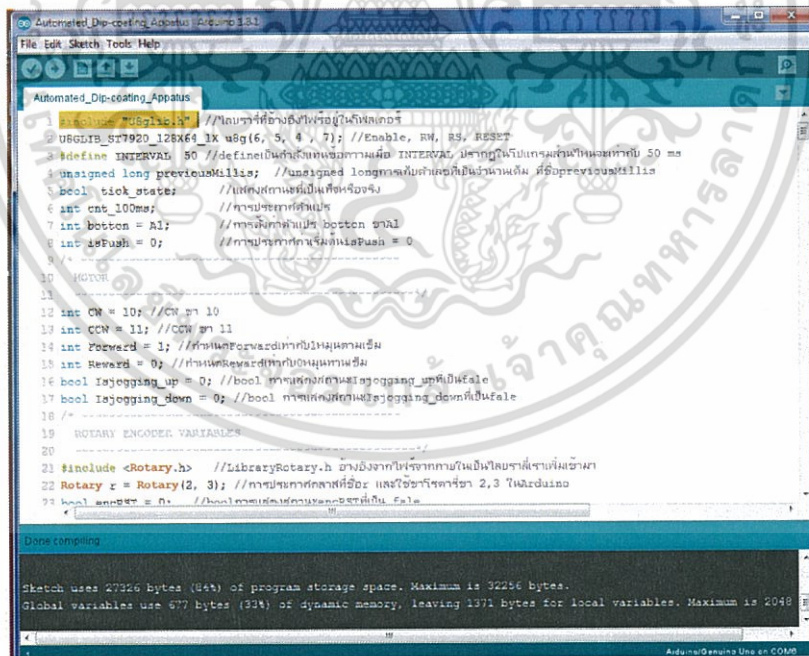


รูปที่ 3 เปิดโปรแกรม Arduino IDE ขึ้นมา แล้วกดไปที่ File>Examples>U8glib เพื่อเช็คได้ว่าได้เพิ่มไลบรารีเข้ามาแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



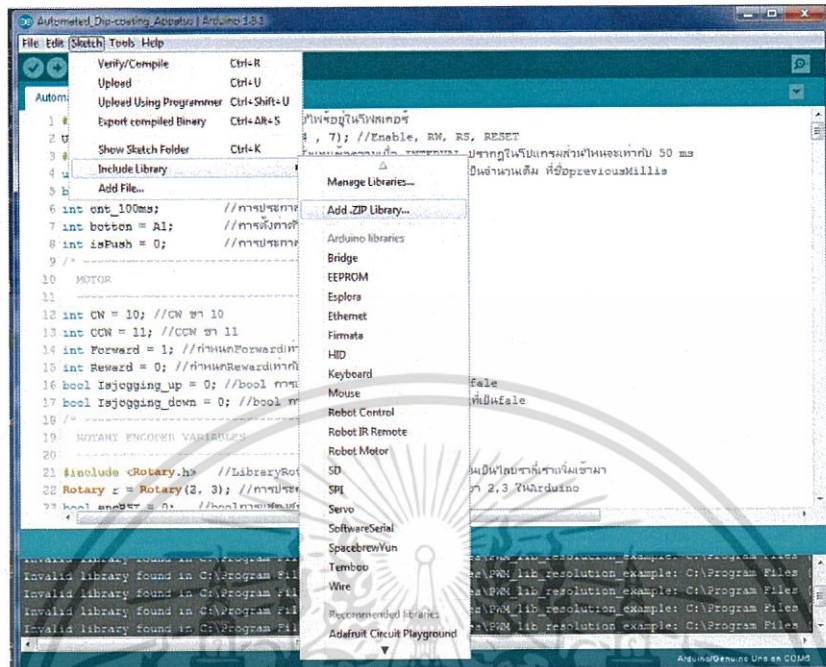
รูปที่ 4 ทดสอบกดปุ่ม Verify ได้โปรแกรม ดูว่ามี error หรือไม่ หากไม่มีแสดงว่า Library ที่ Add เข้าไปใช้งานได้



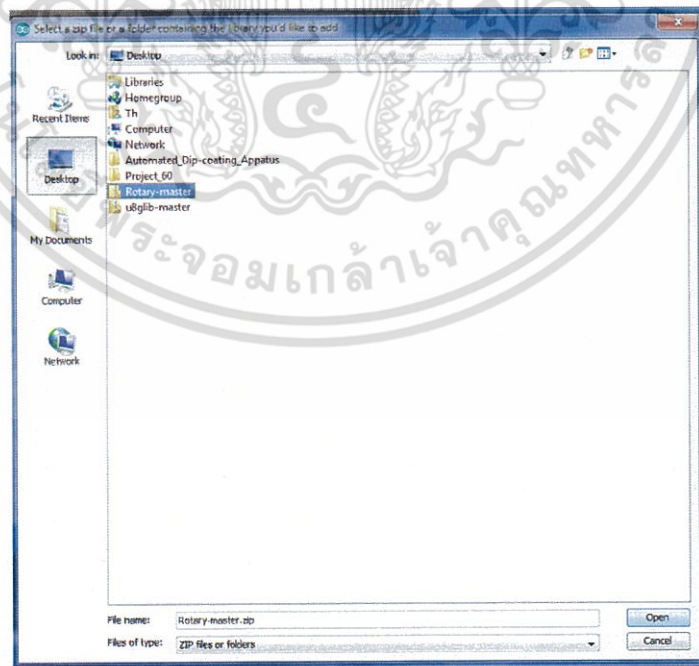
รูปที่ 5 เมื่อ Add Library เรียบร้อยแล้วลอง compiling จะแสดงว่า Done compiling

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.วิธีการลงไลบรารี (Library) สำหรับ Rotary encoder



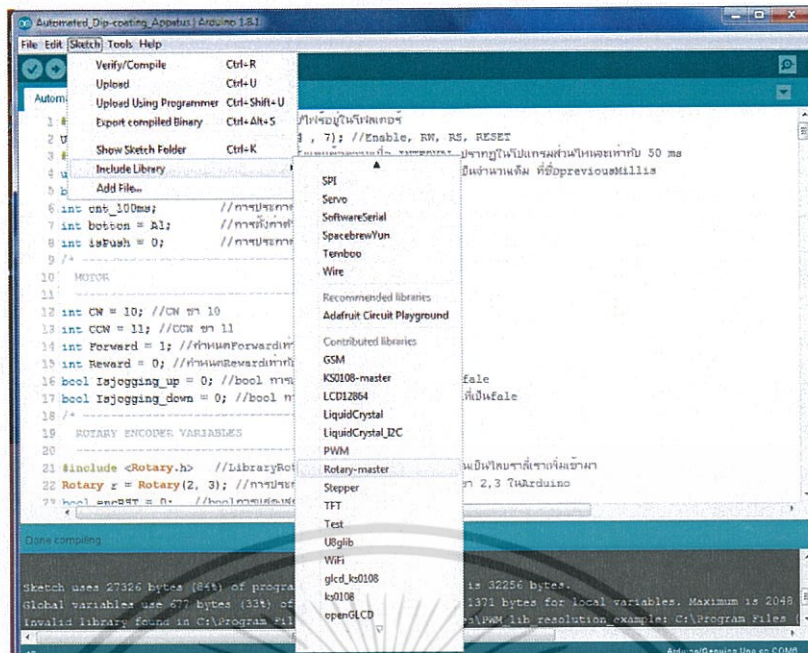
รูปที่ 1 เปิดโปรแกรม Arduino IDE ขึ้นมา แล้วกดไปที่ Sketch > Include Library กดเลือก Include Library > Add ZIP Library



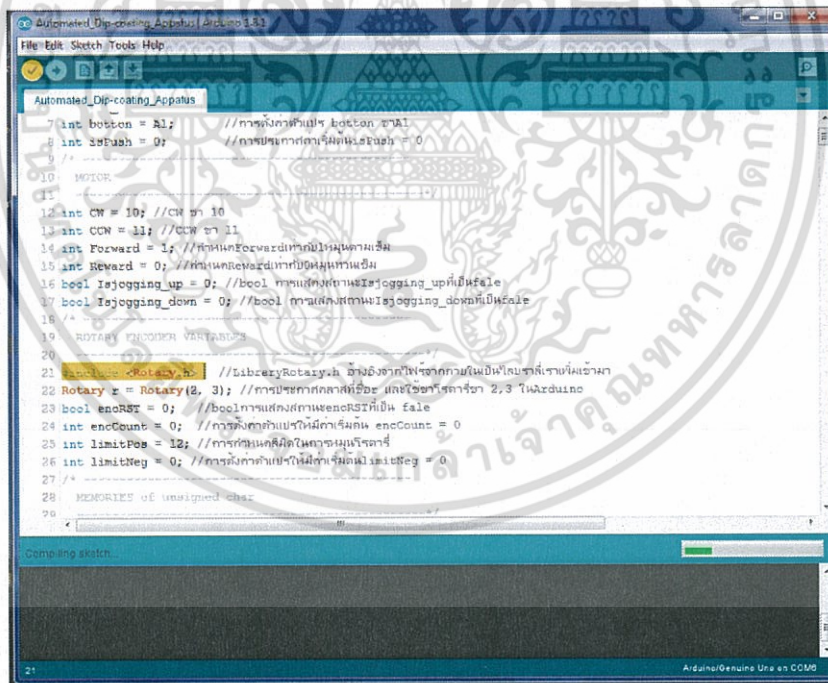
รูปที่ 2 เปิด Folder หรือตำแหน่งที่เก็บ Library เอาไว้ โดยกดเลือกเฉพาะ Folder ของ

Library กด Open

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3 เปิดโปรแกรม Arduino IDE ขึ้นมา แล้วกดไปที่ File>Examples>U8glib เพื่อเช็คว่าได้  
เพิ่มไลบรารีเข้ามาแล้ว



รูปที่ 4 ทดสอบกดปุ่ม Verify ได้โปรแกรม ดูว่ามี error  
หรือไม่ หากไม่มีแสดงว่า Library ที่ Add เข้าไปใช้งานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Automated_Dip-coating_Appatus
7 int botton = A1; //กำหนดค่าปุ่ม botton ว่าA1
8 int isPush = 0; //ค่าของตัวแปรคือisPush = 0
9 /*
10 MOTOR
11 -----*/
12 int CW = 10; //CW ว่า 10
13 int CCW = 11; //CCW ว่า 11
14 int Forward = 1; //กำหนดForwardเท่ากับ1หมายความว่าเริ่ม
15 int Reward = 0; //กำหนดRewardเท่ากับ0หมายความว่าเริ่ม
16 bool Isjogging_up = 0; //bool ค่าแสดงสถานะIsjogging_upที่เป็นfalse
17 bool Isjogging_down = 0; //bool ค่าแสดงสถานะIsjogging_downที่เป็นfalse
18 /*
19 ROTARY ENCODER VARIABLES
20 -----*/
21 #include <Rotary.h> //LibraryRotary.h อ้างอิงจากไลบรารีจากภายในโปรแกรมที่เราเพิ่มเข้ามา
22 Rotary z = Rotary(2, 3); //กำหนดค่าของencoder และใช้ขาที่ขา 2,3 ในArduino
23 bool encRST = 0; //boolค่าแสดงสถานะencRSTที่เป็น false
24 int encCount = 0; //ค่าของตัวแปรคือencCount = 0
25 int limitPos = 12; //ค่าของตัวแปรคือlimitPos
26 int limitNeg = 0; //ค่าของตัวแปรคือlimitNeg = 0
27 /*
28 MEMBERS of unsigned char
29 -----*/
30
Done compiling
Sketch uses 27326 bytes (84% of program storage space. Maximum is 32256 bytes.
Global variables use 677 bytes (33% of dynamic memory, leaving 1371 bytes for local variables. Maximum is 2048
  
```

รูปที่ 5 เมื่อ Add Library เสร็จเรียบร้อยแล้วลอง compiling จะแสดงว่า Done compiling

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ข

### วิธีการใช้เครื่อง Automated Dip-coating Apparatus

1. เปิดสวิตซ์ที่ด้านหลังกล่องควบคุมระบบ
2. หน้าแรกเป็นขั้นตอนในการตั้งค่าข้อมูลโดยใช้ Rotary Encoder
3. เลื่อนแท่นจับ Sample ให้ได้ระยะที่เหมาะสมกับขนาดความสูงของบีกเกอร์โดยหมุน Rotary Encoder ไปที่หน้า Jogging กด ▼ หรือ ▲ เช่นแผ่นรองรับ กว้าง 2.5 cm x 7.5 cm และมีความหนา 1 mm. เมื่อต้องการจุ่มแผ่นรองรับ 2.5 cm สารละลายสูง 4 cm โดยเลื่อนให้แผ่นรองรับอยู่ห่างจากสารละลายในบีกเกอร์ประมาณ 0.5 mm ดังนั้นต้องตั้งค่าระยะในการจุ่ม 30 mm
4. กด Rotary Encoder  $\geq 1$  ครั้งเข้าสู่เมนูหลักเพื่อตั้งค่าต่างๆในการเคลือบ
5. หมุน Rotary Encoder ตามเข็มนาฬิกาหรือทวนเข็มนาฬิกาเพื่อเลือกตำแหน่งที่ต้องการจะตั้งค่า
6. เมื่อเลือกตำแหน่งที่ต้องการ หลังจากนั้น กด Rotary Encoder 1 ครั้งเพื่อแก้ไขตัวเลข
7. กด Rotary Encoder อีก 1 ครั้งเพื่อเลือกค่าที่ต้องการ
8. หลังจากนั้นกด  $\geq$  เพื่อตั้งค่าอื่นๆต่อไป
9. ถ้าหมุนตามเข็มนาฬิกาแล้วกลับมาตำแหน่งแรก ให้ทำการกด 1 ครั้งเพื่อแก้ไขตำแหน่งไหนก็ได้ และหมุนตามเข็มนาฬิกาอีกครั้ง จะสามารถไปยังหน้าที่ต้องการหรือตำแหน่งที่ต้องการได้
10. เมื่อผู้ใช้ตั้งค่าเสร็จเรียบร้อย หมุนเลือกหน้า start จะเข้าสู่กระบวนการทำงาน
11. เมื่อมอเตอร์ทำงาน จอจะแสดงคำว่า RUNNING เมื่อต้องการหยุดมอเตอร์ให้กด Rotary Encoder 1 ครั้ง



ภาคผนวก ค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Program Code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include "U8glib.h" //ไลบรารีที่อ้างอิงไฟร์อยู่ในโฟลเดอร์
U8GLIB_ST7920_128X64_1X u8g(6, 5, 4, 7); //Enable, RW, RS, RESET
#define INTERVAL 50 //defineเป็นคำสั่งแทนข้อความเมื่อ INTERVAL
//ปรากฏในโปรแกรมส่วนไหนจะเท่ากับ 50 ms
unsigned long previousMillis; //unsigned longการเก็บตัวเลขที่เป็นจำนวนเต็ม
//ที่ชื่อpreviousMillis
bool tick_state; //แสดงสถานะที่เป็นเท็จหรือจริง
int cnt_100ms; //การประกาศตัวแปร
int botton = A1; //การตั้งค่าตัวแปร botton ขาA1
int isPush = 0; //การประกาศค่าเริ่มต้นisPush = 0
/* -----
MOTOR
-----*/
int CW = 10; //CW ขา 10
int CCW = 11; //CCW ขา 11
int Forward = 1; //กำหนดForwardเท่ากับ1หมุนตามเข็มนาฬิกา
int Backward = 0; //กำหนดBackwardเท่ากับ0หมุนทวนเข็มนาฬิกา
bool Isjogging_up = 0; //bool การแสดงสถานะ Isjogging_up ที่เป็น
//false
bool Isjogging_down = 0; //bool การแสดงสถานะ Isjogging_downที่เป็น
//false
/* -----
ROTARY ENCODER VARIABLES
-----*/
#include <Rotary.h> //LibraryRotary.h อ้างอิงจากไฟร์จากภายในเป็นไลบรารีเราเพิ่ม
//เข้ามา
Rotary r = Rotary(2, 3); //การประกาศคลาสที่ชื่อ r และใช้ขาโรตารีขา 2,3 ในArduino
bool encRST = 0; //boolการแสดงสถานะencRSTที่เป็น false
int encCount = 0; //การตั้งค่าตัวแปรให้มีค่าเริ่มต้น encCount = 0
int limitPos = 12; //การกำหนดขีดจำกัดในการหมุนโรตารี
int limitNeg = 0; //การตั้งค่าตัวแปรให้มีค่าเริ่มต้นlimitNeg = 0
/* -----
MEMORIES of unsigned char
-----*/

```

unsigned char speed\_U\_N = 0, speed\_U\_F1 = 0, speed\_U\_F2 = 0; นำไปใช้ประโยชน์ด้านการคำนวณว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unsigned char speed_D_N = 0 , speed_D_F1, speed_D_F2 = 0;
unsigned char distance_U_D = 0 , distance_U_N = 0 , distance_U_F1 = 0;
unsigned char distance_D_D = 0 , distance_D_N = 0 , distance_D_F1 = 0;
unsigned char immerstime_T = 0, immerstime_H = 0 , immerstime_D = 0,
immerstime_N = 0;
unsigned char holdingtime_T = 0, holdingtime_H = 0 , holdingtime_D = 0,
holdingtime_N = 0;
unsigned char round_D = 0, round_N = 0 ;
//unsigned char บรรทัดที่ 30-36 เป็นการเก็บข้อมูลอักขระไม่คิดเครื่องหมาย
/* -----
Edit control
-----*/
bool IsEdit = 0; //boolการแสดงสถานะIsEditที่เป็น fale
unsigned char nMenu = 0; //unsigned char เก็บข้อมูลอักขระที่ชื่อMenu(ตำแหน่งและ
แก้ไขแต่ละหลัก)
unsigned char nMode = 0; //unsigned char เก็บข้อมูลอักขระที่ชื่อMode(การหมุน0-9)
unsigned char nPage = 0; //unsigned char เก็บข้อมูลอักขระที่ชื่อPage(เปลี่ยนหน้า)
char buffer[17]; //bufferเพื่อเก็บข้อมูลเป็นBytใน charเก็บตัวอักษร[17]ตัว
bool Isstart = 0; //boolการแสดงสถานะIsstartที่เป็น fale
int nerr = 0; //การตั้งค่าตัวแปรให้มีค่าเริ่มต้นnerr = 0
/* -----
CALCULATE
-----*/
float Speed_up, Speed_down; //floatการเก็บข้อมูลเลขทศนิยมในรูปของสมการ
Speed_up, Speed_down mm/s
float Distance; //floatการเก็บข้อมูลเลขทศนิยมในรูปของสมการ
Distance mm
unsigned long Immerstime; //การประกาศข้อมูลชนิดจำนวนเต็มแบบยาวไม่คิด
เครื่องหมายชื่อImmerstime second
unsigned long Holding; //การประกาศข้อมูลชนิดจำนวนเต็มแบบยาวไม่คิด
เครื่องหมายชื่อHolding second
int dt_up, dt_down; //การประกาศตัวแปรชนิดจำนวนเต็มชื่อdt_up,
dt_down หน่วยms
int pulse_count; //การประกาศตัวแปรชนิดจำนวนเต็มชื่อ pulse_count
int Round, iRound; //การประกาศตัวแปรชนิดจำนวนเต็มชื่อRound, iRound

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unsigned long IM; //การประกาศข้อมูลชนิดจำนวนเต็มแบบยาวไม่คิด
                  //เครื่องหมาย ชื่อ IM
unsigned long HD; //การประกาศข้อมูลชนิดจำนวนเต็มแบบยาวไม่คิด
                  //เครื่องหมาย ชื่อ HD

/* -----
SETUP
----- */
void setup() //โปรแกรมจะทำงานของคำสั่งฟังก์ชันนี้เพียงครั้งเดียว
{
/*-----
MOTOR
----- */
pinMode(CW, OUTPUT); //การกำหนดหน้าที่ของขาส่งสัญญาณCWสัญญาณ
                     OUTPUT
pinMode(CCW, OUTPUT); //การกำหนดหน้าที่ของขาส่งสัญญาณCCWสัญญาณOUTPUT

/*-----
ROTARY ENCODER
----- */
PCICR |= (1 << PCIE2); //บรทัดที่ 72-74 เป็น Intertupชั้นสูง
PCMSK2 |= (1 << PCINT18) | (1 << PCINT19);
sei();
}
/*-----
MAIN LOOP
----- */
void loop() //ฟังก์ชันการวนลูปไปเรื่อยๆ
{
tick(); //ฟังก์ชัน tick
if (tick_state) //tick_state การแสดงสถานะcnt_250ms++
{
cnt_100ms++; //ให้cnt_250ms มีค่าเพิ่มขึ้นทีละ 1
if (cnt_100ms >= 2) //cnt_100ms (Member 2 --> 100ms)
{
cnt_100ms = 0; //เมื่อ cnt_250ms =0

```

เอกสารนี้ isPush = 1; สงวนไว้สำหรับการใช้งาน //แสดงถึงการกดสวิตซ์ isPush = 1 ทำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

isPush = analogRead(botton); //การอ่านค่าการกดสวิตช์ botton
/*-----
Page changing
-----*/
if (isPush == 0 && nMode == 0) //nModeการเลื่อนตำแหน่ง (isPush==0หมายถึงการ
                                หมุน)
{
    switch (nPage) { //การเปลี่ยนหน้า
    case 0: //หน้าแรกP0_1
        switch (nMenu) { //nMenu การเลือกตำแหน่ง
        case 0: //ตำแหน่งที่0
            nPage = 1; //ไปหน้าP1_1(Speed)
            nMenu = 0; //การเก็บตัวอักษรที่ตำแหน่งนี้
            encCount = nMenu; //encCountต้องเท่ากับMenu
            isPush = 1; //isPush เป็น1เมื่อมีการกด
            break;
        }
        break;
    case 1: //speed(P1_1)
        switch (nMenu) {
        case 6: //หมุนไปตำแหน่งที่ 6
            nPage = 2; //ไปหน้าP_2(distance)
            nMenu = 0; //การเก็บตัวอักษรที่ตำแหน่งนี้
            encCount = nMenu;
            isPush = 1; //isPush เป็น1เมื่อมีการกด
            break;
        case 7: //หมุนไปตำแหน่งที่ 7
            nPage = 0; //ไปหน้าP0_1(หน้าแรก)
            nMenu = 0; //การเก็บตัวอักษรที่ตำแหน่งนี้
            encCount = nMenu;
            isPush = 1; //isPush เป็น1เมื่อมีการกด
            break;
        case 8: //หมุนไปตำแหน่งที่ 8
            nPage = 4; //ไปหน้าP2_1(start)
            nMenu = 0; //การเก็บตัวอักษรที่ตำแหน่งนี้
            encCount = nMenu;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    isPush = 1;           //isPush เป็น1เมื่อมีการกด
break;
case 9:                 //หมุนไปตำแหน่งที่ 9
    nPage = 5;         //ไปหน้าP3_1(Jogging)
    nMenu = 0;         //การเก็บตัวอักษรที่ตำแหน่งนี้
    encCount = nMenu;
    isPush = 1;       //isPush เป็น1เมื่อมีการกด
break;
}
break;
case 2:                 //distance(P1_2)
switch (nMenu) {
case 6:                //หมุนไปตำแหน่งที่ 6
    nPage = 3;         //ไปหน้าP1_3(Immerstime)
    nMenu = 0;         //การเก็บตัวอักษรที่ตำแหน่งนี้
    encCount = nMenu;
    isPush = 1;       //isPush เป็น1เมื่อมีการกด
    break;
case 7:                //หมุนไปตำแหน่งที่ 7
    nPage = 1;         //ไปหน้าP1_1(Speed)
    nMenu = 0;         //การเก็บตัวอักษรที่ตำแหน่งนี้
    encCount = nMenu;
    isPush = 1;       //isPush เป็น1เมื่อมีการกด
    break;
case 8:                //หมุนไปตำแหน่งที่ 8
    nPage = 4;         //ไปหน้าP2_1(Start)
    nMenu = 1;         //การเก็บตัวอักษรที่ตำแหน่งนี้
    encCount = nMenu;
    isPush = 1;       //isPush เป็น1เมื่อมีการกด
    break;
case 9:                //หมุนไปตำแหน่งที่ 9
    nPage = 5;         //ไปหน้าP3_1(Jogging)
    nMenu = 0;         //การเก็บตัวอักษรที่ตำแหน่งนี้
    encCount = nMenu;
    isPush = 1;       //isPush เป็น1เมื่อมีการกด
break;
}

```

เอกสารนี้เป็นเอกสารที่วางไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
break;
case 3: //Immerstime(P1_3)
switch (nMenu) {
case 10: //หมุนไปตำแหน่งที่ 8
nPage = 2; //ไปหน้าP1_2(distance)
nMenu = 0; //การเก็บตัวอักษรที่ตำแหน่งนี้
encCount = nMenu;
isPush = 1; //isPush เป็น1เมื่อมีการกด
break;
case 11: //หมุนไปตำแหน่งที่ 9
nPage = 4; //ไปหน้าP2_1(Start)
nMenu = 0; //การเก็บตัวอักษรที่ตำแหน่งนี้
encCount = nMenu;
isPush = 1; //isPush เป็น1เมื่อมีการกด
break;
case 12: //หมุนไปตำแหน่งที่ 10
nPage = 5; //ไปหน้าP3_1(Jogging)
nMenu = 0; //การเก็บตัวอักษรที่ตำแหน่งนี้
encCount = nMenu;
isPush = 1; //isPush เป็น1เมื่อมีการกด
break;
}
break;
case 4: //START (P2_1)
switch (nMenu) {
case 1: //หมุนไปตำแหน่งที่ 1
nPage = 5; //ไปหน้าP3_1(Jogging)
nMenu = 0; //การเก็บตัวอักษรที่ตำแหน่งนี้
encCount = nMenu;
isPush = 1; //isPush เป็น1เมื่อมีการกด
break;
case 2: //หมุนไปตำแหน่งที่ 2
nPage = 1; //ไปหน้าP1_1(speed)
nMenu = 0; //การเก็บตัวอักษรที่ตำแหน่งนี้
encCount = nMenu;

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        isPush = 1;           //isPush เป็น1เมื่อมีการกด
        break;
    }
    break;
case 5:                       //JOGGING(P3_1)
    switch (nMenu) {
        case 2:               //หมุนไปตำแหน่งที่ 2
            nPage = 4;        //ไปหน้าP2_1(Start)
            nMenu = 0;        //การเก็บตัวอักษรที่ตำแหน่งนี้
            encCount = nMenu;
            isPush = 1;       //isPush เป็น1เมื่อมีการกด
            break;
        case 3:               //หมุนไปตำแหน่งที่ 3
            nPage = 1;        //ไปหน้าP1_1(speed)
            nMenu = 0;        //การเก็บตัวอักษรที่ตำแหน่งนี้
            encCount = nMenu;
            isPush = 1;       //isPush เป็น1เมื่อมีการกด
            break;
    }
    break;
}
}
}
//จบPage changing
/*-----*/
shift / edit changing
/*-----*/

if (isPush == 0)              //การหมุน(ถ้าisPush == 1คือการกด)
{
    switch ( nMode) {          //การเลื่อนตำแหน่ง
        case 1:               // การแก้ไข(edit)
            nMode = 0;        //เลื่อนตำแหน่ง
            encCount = nMenu;  //encCountต้องเท่ากับการเลือกตำแหน่งนั้นๆ
            switch (nPage) {  //การเปลี่ยนหน้า
                case 0:        //หน้าแรก (P0_1)
                    limitPos = 0 ; //จำกัดการหมุน 0 ตำแหน่ง
                    limitNeg = 0;
                    break;

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 1:                //หน้าแรก (P1_1)
    limitPos = 9;      //จำกัดการหมุน 9ตำแหน่ง
    limitNeg = 0;
    break;
case 2:                //หน้าแรก (P1_2)
    limitPos = 9;      //จำกัดการหมุน 9ตำแหน่ง
    limitNeg = 0;
    break;
case 3:                //หน้าแรก (P1_3)
    limitPos = 12;     //จำกัดการหมุน 12 ตำแหน่ง
    limitNeg = 0;
    break;
case 4://หน้าแรก (P2_1)
    limitPos = 2;      //จำกัดการหมุน 2 ตำแหน่ง
    limitNeg = 0;
    break;
case 5:                //หน้าแรก (P3_1)
    limitPos = 3;      //จำกัดการหมุน 3 ตำแหน่ง
    limitNeg = 0;
    break;
}
break;
case 0:
    nMode = 1;         //ไม่ใช้การเลื่อนแต่อยู่ตำแหน่งปัจจุบัน
    lsEdit = 1;        //ไม่ใช้การแก้ไขตัวเลข
    break;
}
} //shift / edit changing
/*-----
Edit execution
-----*/
if (lsEdit && nPage == 1) //การแก้ไขตัวเลขในหน้า speed(P1_1)
{
    lsEdit = 0;         //lsEdit=0 แก้ไขตัวเลข
    if (nMenu > 5) nMenu = 0; //nMenu>3 มี3case(nMenu ตำแหน่งที่อยู่แต่ละหลัก)

```

เอกสารนี้เป็น switch (nMenu) สำหรับการใช้งาน //เลื่อนตำแหน่งและแก้ไขแต่ละหลักไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
case 0:
    encCount = speed_U_N; //ตำแหน่งหลักหน่วยของspeed up
    limitPos = 9; //การแก้ไขพารามิเตอร์ 0-9
    limitNeg = 0; //limitNeg เท่ากับ 0
    break;
case 1:
    encCount = speed_U_F1; //ทศนิยมตำแหน่งที่1ของspeed up
    limitPos = 9; //การแก้ไขพารามิเตอร์ 0-9
    limitNeg = 0; //limitNeg เท่ากับ 0
    break;
case 2:
    encCount = speed_U_F2; //ทศนิยมตำแหน่งที่2ของspeed down
    limitPos = 9; //การแก้ไขพารามิเตอร์ 0-9
    limitNeg = 0; //limitNeg เท่ากับ 0
    break;
case 3:
    encCount = speed_D_N; //ตำแหน่งหลักหน่วยของspeed down
    limitPos = 9; //การแก้ไขพารามิเตอร์ 0-9
    limitNeg = 0; //limitNeg เท่ากับ 0
    break;
case 4:
    encCount = speed_D_F1; //ทศนิยมตำแหน่งที่1ของspeed down
    limitPos = 9; //การแก้ไขพารามิเตอร์ 0-9
    limitNeg = 0; //limitNeg เท่ากับ 0
    break;
case 5:
    encCount = speed_D_F2; //ทศนิยมตำแหน่งที่2ของspeed down
    limitPos = 9; //การแก้ไขพารามิเตอร์ 0-9
    limitNeg = 0; //limitNeg เท่ากับ 0
    break;
}
} //การแก้ไขตัวเลขในหน้า speed(P1_1)
if (IsEdit && nPage == 2) //การแก้ไขตัวเลขในหน้า distance(P1_2)
{

```

เอกสารนี้เป็น IsEdit=0; //IsEdit=0 แก้ไขตัวเลข เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (nMenu > 5) nMenu = 0; //5 case(ตำแหน่งที่อยู่แต่ละหลัก)
switch (nMenu)
{
case 0:
    encCount = distance_U_D; //ตำแหน่งหลักสิบของdistance up
    limitPos = 9; //การแก้ไขพารามิเตอร์ 0-9
    limitNeg = 0; //limitNeg เท่ากับ 0
    break;
case 1:
    encCount = distance_U_N; //ตำแหน่งหลักหน่วยของdistance up
    limitPos = 9; //การแก้ไขพารามิเตอร์ 0-9
    limitNeg = 0; //limitNeg เท่ากับ 0
    break;
case 2:
    encCount = distance_U_F1; //ทศนิยมตำแหน่งที่1ของdistance up
    limitPos = 9; //การแก้ไขพารามิเตอร์ 0-9
    limitNeg = 0; //limitNeg เท่ากับ 0
    break;
case 3:
    encCount = distance_D_D; //ตำแหน่งหลักสิบของdistance down
    limitPos = 9; //การแก้ไขพารามิเตอร์ 0-9
    limitNeg = 0; //limitNeg เท่ากับ 0
    break;
case 4:
    encCount = distance_D_N; //ตำแหน่งหลักหน่วยของdistance down
    limitPos = 9; //การแก้ไขพารามิเตอร์ 0-9
    limitNeg = 0; //limitNeg เท่ากับ 0
    break;
case 5:
    encCount = distance_D_F1; //ทศนิยมตำแหน่งที่1ของdistance down
    limitPos = 9; //การแก้ไขพารามิเตอร์ 0-9
    limitNeg = 0; //limitNeg เท่ากับ 0
    break;
}
} //การแก้ไขตัวเลขในหน้าdistance(P1_2)

```

เอกสารนี้ if (IsEdit && nPage == 3) การใช้งานเพื่อการคิด //การแก้ไขตัวเลขในหน้า Immerstime(P1\_3) การคำนวณค่า  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
IsEdit = 0; //IsEdit=0 แก้ไขตัวเลข
if (nMenu > 9) nMenu = 0; //7 case(nMenu ตำแหน่งที่อยู่แต่ละหลัก)
switch (nMenu)
{
case 0:
encCount = immersetime_T; //ตำแหน่งหลักร้อยของ immersetime
limitPos = 9; //การแก้ไขพารามิเตอร์ 0-9
limitNeg = 0; //limitNeg เท่ากับ 0
break;
case 1:
encCount = immersetime_H; //ตำแหน่งหลักสิบของ immersetime
limitPos = 9; //การแก้ไขพารามิเตอร์ 0-9
limitNeg = 0; //limitNeg เท่ากับ 0
break;
case 2:
encCount = immersetime_D; //ตำแหน่งหลักหน่วยของ immersetime
limitPos = 9; //การแก้ไขพารามิเตอร์ 0-9
limitNeg = 0; //limitNeg เท่ากับ 0
break;
case 3:
encCount = immersetime_N; //ตำแหน่งหลักหน่วยของ immersetime
limitPos = 9; //การแก้ไขพารามิเตอร์ 0-9
limitNeg = 0; //limitNeg เท่ากับ 0
break;
case 4:
encCount = holdingtime_T; //ตำแหน่งหลักร้อยของ holdingtime
limitPos = 9; //การแก้ไขพารามิเตอร์ 0-9
limitNeg = 0; //limitNeg เท่ากับ 0
break;
case 5:
encCount = holdingtime_H; //ตำแหน่งหลักสิบของ holdingtime
limitPos = 9; //การแก้ไขพารามิเตอร์ 0-9
limitNeg = 0; //limitNeg เท่ากับ 0
break;
case 6:
encCount = holdingtime_D; //ตำแหน่งหลักหน่วยของ holdingtime
limitPos = 9; //การแก้ไขพารามิเตอร์ 0-9
limitNeg = 0; //limitNeg เท่ากับ 0
break;
}
}

```

เอกสารนี้เป็น case 6: ที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    encCount = holdingtime_D;           //ตำแหน่งหลักหน่วยของ holdingtime
    limitPos = 9;                       //การแก้ไขพารามิเตอร์ 0-9
    limitNeg = 0;                       //limitNeg เท่ากับ 0
    break;
case 7:
    encCount = holdingtime_N;          //ตำแหน่งหลักหน่วยของ holdingtime
    limitPos = 9;                       //การแก้ไขพารามิเตอร์ 0-9
    limitNeg = 0;                       //limitNeg เท่ากับ 0
    break;
case 8: //round_D
    encCount = round_D ;               //ตำแหน่งหลักสิบของround
    limitPos = 9;                       //การแก้ไขพารามิเตอร์ 0-9
    limitNeg = 0;                       //limitNeg เท่ากับ 0
    break;
case 9:
    encCount = round_N ;               //ตำแหน่งหลักหน่วยของround
    limitPos = 9;                       //การแก้ไขพารามิเตอร์ 0-9
    limitNeg = 0;                       //limitNeg เท่ากับ 0
    break;
}
}
//=====
//      START PAGE & CALCULATION
//=====
if (IsEdit && nPage == 4) {            //สมการที่ทำให้มอเตอร์เคลื่อนที่ในหน้า
    Start(P2_1)
    IsEdit = 0;                        //IsEdit=0 แก้ไขตัวเลข
    if (nMenu == 0 && Isstart == 0)    //nMenuและIsstart== 0
    {
        Isstart = 1;                  //Isstart เป็นจริงจะทำให้nerr = 0เกิดError
        nerr = 0;                      //ถ้าไม่ตั้งค่าข้อมูลใดข้อมูลหนึ่งเมื่อกดstart จะ
                                        //ขึ้นError
                                        // ณ ตอนนี้อย่างไม่มี การตรวจสอบ error

```

เอกสารนี้เป็น //คำนวณ speed up; mm/s งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Speed_up = (float)speed_U_N + 0.1 * ((float)speed_U_F1) + 0.1 * ((float)speed_U_F2);
//สมการความเร็วตั้งขึ้นเป็นทศนิยมต้องใส่(float)หน้าตำแหน่งนั้น
    //คำนวณspeed down , mm/s
Speed_down = (float)speed_D_N + 0.1 * ((float)speed_D_F1) + 0.1 *
((float)speed_D_F2);
//สมการความเร็วจุ่มลงเป็นทศนิยมต้องใส่(float)หน้าตำแหน่งนั้น
    //คำนวณdistance up , mm
Distance = (float)distance_U_D * 10 + (float)distance_U_N + 0.1 *
((float)distance_U_F1);
//สมการระยะทางเป็นเลขจำนวนเต็มจึงไม่ใส่(float)
    //คำนวณ Immersetime
Immersetime = immersetime_T * 1000 + immersetime_H * 100 + immersetime_D * 10
+ immersetime_N; //สมการเวลาแช่เป็นเลขจำนวนเต็มจึงไม่ใส่(float)
IM = Immersetime * 1000; //สร้างสมการขึ้นมาเพื่อหน่วยเวลาของImmersetimeใน
หน่วย(ms)
    //คำนวณ holding
Holding = holdingtime_T * 1000 + holdingtime_H * 100 + holdingtime_D * 10 +
holdingtime_N; //สมการเวลาพักเป็นเลขจำนวนเต็มจึงไม่ใส่(float)
HD = Holding * 1000; //สร้างสมการขึ้นมาเพื่อหน่วยเวลาของ Holdingในหน่วย
(ms)

    //คำนวณ round
Round = round_D * 10 + round_N; //สมการจำนวนรอบในการจุ่ม
    // จำนวน pulse ของ distance
pulse_count = (int)(Distance / 0.002); //สมการคำนวณ pulse จากระยะทาง/2ไมครอน
(1step)

    // เวลา dt แต่ละสเต็ป ขาขึ้น ( 0.002mm*1000ms/s)/(speed_up:mm/s)
dt_up = 1000000 / (Speed_up * 500); //สมการdelayMicroseconds [1
Microseconds/(Speed_up * 1 สเต็ป)

    // เวลา dt แต่ละสเต็ป ขาลง
dt_down = 1000000 / (Speed_down * 500); //สมการdelayMicroseconds [1
Microseconds/(Speed_down* 1 สเต็ป)]
}
} //จบสมการที่ทำให้มอเตอร์เคลื่อนที่ในหน้า Start(P2_1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//=====
// JOGGING PAGE
//=====

if (IsEdit && nPage == 5) { //การเคลื่อนของมอเตอร์ขึ้น-ลงโดยการกดสวิทช์ค้างใน
                            //หน้า Jogging(P3_1)
    IsEdit = 0; //IsEdit=0 แก้ไขตัวเลข
    if (nMenu == 0 && Isjogging_up == 0) //ถ้านMenu == 0 การเลือกตำแหน่งและ
                                        //Isjogging_up เป็นเท็จ
    {
        Isjogging_up = 1; //เงื่อนไข Isjogging_up เป็นจริง
    }
    if (nMenu == 1 && Isjogging_down == 0) //ถ้านMenu == 1 ไม่ได้เลือกตำแหน่งและ
                                        //Isjogging_up เป็นเท็จ
    {
        Isjogging_down = 1; //เงื่อนไข Isjogging_down เป็นจริง
    }
} //จบการกดสวิทช์ค้างในหน้า Jogging(P3_1)
// Update encCount to parameters
/*-----
ROTARY ENCODER
-----*/
if (nMode == 0) nMenu = encCount; //การเลื่อนตำแหน่ง nMode == 0 เมื่อ nMenu
                                  // = encCount
if (nMode == 1 && nPage == 1) //mode 1 การshift หน้าP1_1
{
    switch (nMenu) //เลือกตำแหน่ง
    {
        case 0: // speed_N
            speed_U_N = encCount; //speed_U_N ต้องเท่ากับ encCount (ใช้ในการ
                                  //เลือกตำแหน่งที่ต้องการแก้ไข)

            break;
        case 1:
            speed_U_F1 = encCount; //speed_U_F1 ต้องเท่ากับ encCount (ใช้ใน
                                  //การเลือกตำแหน่งที่ต้องการแก้ไข)

            break;
    }
}

```

เอกสารนี้เป็น case 2: ที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

speed_U_F2 = encCount; //speed_D_N ต้องเท่ากับ encCount (ใช้ในการ
                        //เลือกตำแหน่งที่ต้องการแก้ไข)

break;
case 3:
speed_D_N = encCount; //speed_D_F1 ต้องเท่ากับ encCount (ใช้ใน
                        //การเลือกตำแหน่งที่ต้องการแก้ไข)

break;
case 4:
speed_D_F1 = encCount; //speed_D_F1 ต้องเท่ากับ encCount (ใช้ใน
                        //การเลือกตำแหน่งที่ต้องการแก้ไข)

break;
case 5:
speed_D_F2 = encCount; //speed_D_F1 ต้องเท่ากับ encCount (ใช้ใน
                        //การเลือกตำแหน่งที่ต้องการแก้ไข)

break;
}
} //mode 1 การshift หน้าP1_1
if (nMode == 0) nMenu = encCount; //การเลื่อนตำแหน่งnMode == 0 เมื่อ nMenu
= encCount

if (nMode == 1 && nPage == 2) //mode 1 การshift หน้าP1_2
{
switch (nMenu) //เลือกตำแหน่ง
{
case 0: //distance_N
distance_U_D = encCount; //distance_U_D ต้องเท่ากับ encCount (ใช้ใน
                           //การเลือกตำแหน่งที่ต้องการแก้ไข)

break;
case 1:
distance_U_N = encCount; //distance_U_N ต้องเท่ากับ encCount (ใช้ใน
                           //การเลือกตำแหน่งที่ต้องการแก้ไข)

break;
case 2:
distance_U_F1 = encCount; //distance_U_F1 ต้องเท่ากับ encCount (ใช้ใน
                           //การเลือกตำแหน่งที่ต้องการแก้ไข)

```

เอกสารนี้เป็น break; สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 3:
    distance_D_D = encCount;           //distance_D_D ต้องเท่ากับ encCount (ใช้ในการเลือกตำแหน่งที่ต้องการแก้ไข)

    break;
case 4:
    distance_D_N = encCount;           //distance_D_N ต้องเท่ากับ encCount (ใช้ในการเลือกตำแหน่งที่ต้องการแก้ไข)

    break;
case 5:
    distance_D_F1 = encCount;          //distance_D_F1 ต้องเท่ากับ encCount (ใช้ในการเลือกตำแหน่งที่ต้องการแก้ไข)

    break;
}
}
if (nMode == 0) nMenu = encCount;     // การหมุนเลข 0-9
if (nMode == 1 && nPage == 3)        //mode 1 การshift หน้าP1_2
{
    switch (nMenu)                    //เลือกตำแหน่ง
    {
        case 0:
            immersetime_T = encCount; //immersetime_H ต้องเท่ากับ encCount (ใช้ในการเลือกตำแหน่งที่ต้องการแก้ไข)

            break;
        case 1:
            immersetime_H = encCount; //immersetime_H ต้องเท่ากับ encCount (ใช้ในการเลือกตำแหน่งที่ต้องการแก้ไข)

            break;
        case 2:
            immersetime_D = encCount; //immersetime_D ต้องเท่ากับ encCount (ใช้ในการเลือกตำแหน่งที่ต้องการแก้ไข)

            break;
        case 3:
            immersetime_N = encCount; //immersetime_N ต้องเท่ากับ encCount (ใช้ในการเลือกตำแหน่งที่ต้องการแก้ไข)

            break;

```

เอกสารนี้เป็น case 4: ที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    holdingtime_T = encCount; //holdingtime_H ต้องเท่ากับ encCount (ใช้ในการเลือกตำแหน่งที่ต้องการแก้ไข)

    break;
case 5:
    holdingtime_H = encCount; // holdingtime_D ต้องเท่ากับ encCount (ใช้ในการเลือกตำแหน่งที่ต้องการแก้ไข)

    break;
case 6:
    holdingtime_D = encCount; //holdingtime_N ต้องเท่ากับ encCount (ใช้ในการเลือกตำแหน่งที่ต้องการแก้ไข)

    break;
case 7:
    holdingtime_N = encCount; //holdingtime_N ต้องเท่ากับ encCount (ใช้ในการเลือกตำแหน่งที่ต้องการแก้ไข)

    break;
case 8:
    round_D = encCount; //round_D ต้องเท่ากับ encCount (ใช้ในการเลือกตำแหน่งที่ต้องการแก้ไข)

    break;
case 9:
    round_N = encCount; // round_N ต้องเท่ากับ encCount (ใช้ในการเลือกตำแหน่งที่ต้องการแก้ไข)

    break;
}
} //mode 1 การshift หน้าP1_2
// -----
switch (nPage) { //การเปลี่ยนหน้าโดยทำงานในแต่ละ case
case 0: //เป็นการปรีนตัวอักษรโดยการdraw_P0_1()ของไลบรารีu8g

    u8g.firstPage();
    do
    {
        draw_P0_1(); //หน้าแรก(P0_1)
    }
    while ( u8g.nextPage() );
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 1:
    u8g.firstPage();
    do
    {
        draw_P1_1();                //หน้าspeed(P1_1)
    }
    while ( u8g.nextPage() );
    break;
case 2:
    u8g.firstPage();
    do
    {
        draw_P1_2();                //หน้าdistance(P1_2)
    }
    while ( u8g.nextPage() );
    break;
case 3:
    u8g.firstPage();
    do
    {
        draw_P1_3();                //หน้าImmerstime(P1_3)
    }
    while ( u8g.nextPage() );
    break;
case 4:
    u8g.firstPage();
    do
    {
        draw_P2_1();                //หน้าstart(P2_1)
    }
    while ( u8g.nextPage() );
    break;
case 5:
    u8g.firstPage();
    do

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        draw_P3_1();                                //หน้าjogging(P3_1)
    }
    while ( u8g.nextPage() );
    break;
}
} // จบcnt_100ms
} // จบtick_state
//=====
//=====
// DEEP COATING START
//=====
//=====
if (Isstart) {
    // ตรวจสอบการคำนวณ
    // มีความผิดพลาด
    if (nerr < 0)
    {
        delay(3000); // รอแสดงผล
    }
    else // ไม่ผิดพลาด
    {
        // เริ่มส่งมอเตอร์
        IsFinished = 0;
        for (iRound = 0; iRound < Round; iRound++) // จำนวนรอบในการจุ่ม
        {
            FORWARD(); // มอเตอร์หมุนตามเข็มนาฬิกา(การจุ่มลง)
            if (isPush == 0) break; // หยุดด้วยการกด
            delay(IM); // delay(ms)
            if (isPush == 0) break; // หยุดด้วยการกด
            BACKWARD(); // มอเตอร์หมุนทวนเข็มนาฬิกา(การดึงขึ้น)
            delay(HD); // delay(ms)
            if (isPush == 0) break; // หยุดด้วยการกด
        } // จบfor(iRound)
    } // จบ else (เงื่อนไขที่ไม่มีข้อผิดพลาด)
}
// รีเซตการเริ่มใหม่
IsFinished = 0;
Isstart = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการวิจัยและพัฒนาเท่านั้น ไม่ควรนำเอกสารนี้ไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

IsEdit = 0; //ไม่ได้แก้ไขตัวเลข(ถ้าแก้ไขจะเท่ากับ1)
} //จบif(Isstart)

//=====
// JOGING
//=====
if (Isjogging_up) //สั่งjogging_up
{
do
{
digitalWrite(CCW, Backward); //CCW การกำหนดทิศทางของมอเตอร์เท่ากับ
//การทวนเข็มนาฬิกา

digitalWrite(CW, HIGH); //การส่งค่าลอจิก CW ที่เป็น HIGH
delayMicroseconds(150); //delayMicroseconds
digitalWrite(CW, LOW); //การส่งค่าลอจิก CW ที่เป็น LOW
delayMicroseconds(150); //delayMicroseconds
isPush = analogRead(botton); //การอ่านค่าการกดbottonที่ขาA1
} while (isPush == 0); //whileทำงานต่อจากdo
digitalWrite(CW, HIGH); //การส่งค่าลอจิก CW ที่เป็น LOW(เมื่อปล่อย
//สวิตช์)

Isjogging_up = 0; //
IsEdit = 0; //ไม่ได้แก้ไขตัวเลข(ถ้าแก้ไขจะเท่ากับ1)
} //จบIsjogging_up
if (Isjogging_down) //สั่งjogging_down
{
do
{
digitalWrite(CCW, Forward); //CCW การกำหนดทิศทางของมอเตอร์เท่ากับตามเข็
//นาฬิกา

digitalWrite(CW, HIGH); //การส่งค่าลอจิก CW ที่เป็น HIGH
delayMicroseconds(150); //delayMicroseconds
digitalWrite(CW, LOW); //การส่งค่าลอจิก CW ที่เป็น LOW
delayMicroseconds(150); //delayMicroseconds
isPush = analogRead(botton); //การอ่านค่าการกดbottonที่ขาA1
} while (isPush == 0); //whileทำงานต่อจากdo
digitalWrite(CW, HIGH); //การส่งค่าลอจิก CW ที่เป็น LOW(เมื่อปล่อย
//สวิตช์)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    Isjogging_down = 0;
    IsEdit = 0;
}    // if Isjogging_down
}    //end loop

/* -----
      MOTOR UP-DOWN
-----*/

void BACKWARD()                //DOWN
{
    digitalWrite(CCW, Backward);    //CCW การกำหนดทิศทางของมอเตอร์เท่ากับ
                                     การตามเข็มนาฬิกา
    for (int a = 0; a <= pulse_count; a++) //เมื่อมีค่าเริ่มต้นเท่ากับ0 ถ้าน้อยกว่าหรือ
                                     เท่ากับpulse_count a จะเพิ่มขึ้นทีละ 1จนถึง
                                     pulse_count
    {
        digitalWrite(CW, HIGH);    //การส่งค่าลอจิก CW ที่เป็น HIGH
        delayMicroseconds(dt_up / 2); //dt_up/2 คือคาบเวลาหาร2จะได้
        delayMicroseconds(dt_down / 2); //dt_down/2 คือคาบเวลาหาร2จะได้
        digitalWrite(CW, LOW);    //การส่งค่าลอจิก CW ที่เป็น LOW
        delayMicroseconds(dt_up / 2); //dt_up/2 คือคาบเวลาหาร2จะได้
        delayMicroseconds(dt_down / 2); //dt_down/2 คือคาบเวลาหาร2จะได้
        isPush = analogRead(botton); //การอ่านค่าการกดbottonที่ขาA1
        if (isPush == 0) break; // หยุดด้วยการกด
    }
}

void FORWARD()                //UP
{
    digitalWrite(CCW, Backward);    //CCW การกำหนดทิศทางของมอเตอร์เท่ากับ
                                     การทวนเข็มนาฬิกา
    for (int a = 0; a <= pulse_count; a++) //เมื่อมีค่าเริ่มต้นเท่ากับ0 ถ้าน้อยกว่าหรือ
                                     เท่ากับpulse_count a จะเพิ่มขึ้นทีละ 1จนถึง
                                     pulse_count
    {
        digitalWrite(CW, HIGH);    //การส่งค่าลอจิก CW ที่เป็น HIGH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

delayMicroseconds(dt_down / 2); //dt_down/2 คือคาบเวลาหาร2จะได้
delayMicrosecondsที่เป็น HIGH
digitalWrite(CW, LOW); //การส่งค่าลอจิก CW ที่เป็น LOW
delayMicroseconds(dt_down / 2); //dt_down/2 คือคาบเวลาหาร2จะได้
delayMicrosecondsที่เป็น LOW
    if (isPush == 0) break; // หยุดด้วยการกด
}
} //จบ REWARD
/* -----

```

## FUNCTIONS

```

-----*/
void tick() //เป็นการใช้tickแทนdelay
{
    tick_state = false; //tick_state เป็นเท็จ
    if (millis() - previousMillis > INTERVAL)
    {
        previousMillis = millis(); //เมื่อpreviousMillisเท่ากับmillis
        tick_state = true; //สถานะtick_stateจะเป็นจริง
    }
} //จบvoid tick
/* -----

```

## ROTARY ENCODER INTERRUPTION

```

-----*/
ISR(PCINT2_vect) { //Interrupt service routine(ISR)ทิศทางการ
    //ทำงานของINTERRUP

    unsigned char result = r.process();
    // Reset encoder counter
    if (encRST == 1) encCount = 0; //
    if (result == DIR_NONE){ //ทำให้resultที่ชื่อDIR_NONEไม่ทำอะไร
        // do nothing
    }
    else if (result == DIR_CW) //resultจะเก็บอักษรชื่อ DIR_CW หมุนตามเข็มนาฬิกา
    {
        encCount++; // encCountจะเพิ่มทีละ 1 ไปเรื่อยๆ
        if (encCount > limitPos) encCount = limitNeg; //เมื่อencCountเพิ่มค่าทีละ1จนมากกว่า

```

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี การนำเอกสารนี้ไปใช้ในการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

limitPos ให้ encCount = limitNeg คือ
encCountจะเท่ากับ0

}
else if (result == DIR_CCW) //resultจะเก็บอักษรชื่อ DIR_CCW หมุนทวนเข็มนาฬิกา
{
encCount--; // encCountจะลดทีละ 1 ไปเรื่อยๆ
if (encCount < limitNeg) encCount = limitPos; //เมื่อencCountลดค่าทีละ1จนน้อยกว่า
limitNeg จะทำให้ encCount =
limitNeg คือencCountจะเท่ากับ9
}
}
void draw_P0_1(void)
{
//dip-coating
u8g.drawRFrame(0, 0, 128, 64, 0); //frame big
u8g.setFont(u8g_font_helvR08);
u8g.drawStr(3, 14, "1.Push the botton to edit.");
u8g.drawStr(3, 27, "2.Rotate to next digit.");
u8g.drawStr(3, 40, "3.Press the botton to");
u8g.drawStr(3, 53, "enter the main menu.");
u8g.setFont(u8g_font_timR14); //next
u8g.drawStr(112, 60, ">");
/* -----
แสดงการกด
-----*/
if (lisPush) //if..else คำสั่งการตัดสินใจ
u8g.drawStr(10, 25, " "); //แสดงการกดถ้าlisPushไม่เป็น0แต่เท่ากับ1
else
u8g.drawStr(10, 25, " "); //หรือการปล่อยสวิตซ์

switch (nMode) { //การเลื่อน
case 0: //shift
switch (nMenu) { //เส้นใต้แต่ละตำแหน่ง
case 0: //next
u8g.drawLine(110, 60, 117, 60); //เส้นใต้ >

```

เอกสารนี้เป็นลิขสิทธิ์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        break;
    }
    break;
}
}
void draw_P1_1(void)
{
    //dip-coating
    u8g.drawRFrame(0, 0, 128, 64, 0);          //frame big
    u8g.setFont(u8g_font_baby);
    u8g.drawStr(4, 8, "Dip-Coat");
    u8g.drawFrame(40, 0, 124, 14);
    u8g.drawStr(49, 8, "Start");
    u8g.drawFrame(80, 0, 124, 14);          //frame jogging
    u8g.drawStr(88, 8, "Jogging");
    //menu set
    u8g.setFont(u8g_font_6x12);
    /* -----
       แสดงการกด
    -----*/
    if (lisPush)
        u8g.drawStr(10, 25, " ");
    else
        u8g.drawStr(10, 25, " ");
    /* -----
       speed_up
    -----*/

    u8g.drawStr(10, 25, "Speed:");
    u8g.drawStr(10, 40, "down=");
    sprintf(buffer, "%d", speed_U_N);
    u8g.drawStr(42, 40, buffer);
    u8g.drawStr(50, 40, ".");          //ทศนิยม
    sprintf(buffer, "%d", speed_U_F1);
    u8g.drawStr(56, 40, buffer);
    sprintf(buffer, "%d", speed_U_F2);
    u8g.drawStr(64, 40, buffer);
}

```

เอกสารนี้จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

u8g.drawStr(73, 40, "mm/s");
/*-----
    speed_down
-----
*/
u8g.drawStr( 10, 52, "up=");
sprintf(buffer, "%d", speed_D_N);
u8g.drawStr( 30, 52 , buffer);
u8g.drawStr(38, 52, ".");
sprintf(buffer, "%d", speed_D_F1);
u8g.drawStr( 44, 52 , buffer);
sprintf(buffer, "%d", speed_D_F2);
u8g.drawStr( 52, 52 , buffer);
u8g.drawStr(61, 52, "mm/s");
u8g.setFont(u8g_font_timR14); //next
u8g.drawStr(112, 60, ">");
u8g.drawStr(90, 60, "<"); //back
switch (nMode) {
  case 0: //shift
    switch (nMenu) { //เส้นใต้แต่ละตำแหน่ง
      case 0:
        u8g.drawLine(42, 40, 47, 40); //speed_D_N
        break;
      case 1:
        u8g.drawLine(56, 40, 61, 40); //speed_D_F1
        break;
      case 2:
        u8g.drawLine(64, 40, 69, 40); //speed_D_F2
        break;
      case 3:
        u8g.drawLine(30, 52, 35, 52); //speed_U_N
        break;
      case 4:
        u8g.drawLine(44, 52, 49, 52); //speed_U_F1
        break;

```

เอกสารนี้เป็น case 5: รที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    u8g.drawLine(52, 52, 57, 52); //speed_U_F2
    break;
case 6: //next
    u8g.drawLine(110, 60, 117, 60);
    break;
case 7: //back
    u8g.drawLine(95, 60, 102, 60);
    break;
case 8:
    u8g.drawLine(49, 9, 68, 9); //start
    break;
case 9:
    u8g.drawLine(88, 11, 118, 11); //jogging
    break;
}
break;
case 1: //edit
switch (nMenu) {
case 0:
    u8g.drawFrame(40, 31, 9, 11); // speed_D_N
    break;
case 1:
    u8g.drawFrame(54, 31, 9, 11); //speed_D_f1
    break;
case 2:
    u8g.drawFrame(62, 31, 9, 11); //speed_D_f2
    break;
case 3:
    u8g.drawFrame(28, 43, 9, 11); //speed_U_N
    break;
case 4:
    u8g.drawFrame(42, 43, 9, 11); //speed_U_f1
    break;
case 5:
    u8g.drawFrame(50, 43, 9, 11); //speed_U_f2

```

เอกสารนี้เป็น **break**; ที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    break;
}
}
void draw_P1_2(void)
{
    //dip-coating
    u8g.drawRFrame(0, 0, 128, 64, 0);
    u8g.setFont(u8g_font_baby);
    u8g.drawStr(4, 8, "Dip-Coat");
    u8g.drawFrame(40, 0, 124, 14);
    u8g.drawStr(49, 8, "Start");
    u8g.drawFrame(80, 0, 124, 14);
    u8g.drawStr(88, 8, "Jogging");

    //menu set
    u8g.setFont(u8g_font_6x12);
    /* -----
       แสดงการกด
    ----- */
    if (lisPush)
        u8g.drawStr(10, 25, " ");
    else
        u8g.drawStr(10, 25, " ");
    /* -----
       distance_up
    ----- */
    u8g.drawStr(10, 25, "Distance:");
    u8g.drawStr(10, 40, "down=");
    sprintf(buffer, "%d", distance_U_D);
    u8g.drawStr (42, 40 , buffer);
    sprintf(buffer, "%d", distance_U_N);
    u8g.drawStr (50, 40 , buffer);
    u8g.drawStr(58, 40, ".");    //ทศนิยม
    sprintf(buffer, "%d", distance_U_F1);
    u8g.drawStr (64, 40 , buffer);

```

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
 การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

u8g.drawStr(73, 40, "mm");
/*-----
    distance_down
-----*/

u8g.drawStr( 10, 52, "up=");
sprintf(buffer, "%d", distance_D_D);
u8g.drawStr( 30, 52 , buffer);
sprintf(buffer, "%d", distance_D_N);
u8g.drawStr( 38, 52 , buffer);
u8g.drawStr(46, 52, "."); //ทศนิยม
sprintf(buffer, "%d", distance_D_F1);
u8g.drawStr( 52, 52 , buffer);
u8g.drawStr(61, 52, "mm");
u8g.setFont(u8g_font_timR14); //next1
u8g.drawStr(112, 60, ">"); //next
u8g.drawStr(90, 60, "<"); //back
switch (nMode) {
case 0://shift
switch (nMenu) { //เส้นใต้แต่ละตำแหน่ง
case 0:
u8g.drawLine(42, 40, 47, 40); // distance_D_D
break;
case 1:
u8g.drawLine(50, 40, 55, 40); // distance_D_N
break;
case 2:
u8g.drawLine(64, 40, 69, 40); // distance_D_F1
break;
case 3:
u8g.drawLine(30, 52, 35, 52); // distance_U_D
break;
case 4:
u8g.drawLine(38, 52, 43, 52); // distance_U_N
break;
case 5:
u8g.drawLine(52, 52, 57, 52); // distance_U_F1

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    break;
case 6: //next
    u8g.drawLine(110, 60, 117, 60);
    break;
case 7: //back
    u8g.drawLine(95, 60, 102, 60);
    break;
case 8:
    u8g.drawLine(49, 9, 68, 9); //start
    break;
case 9:
    u8g.drawLine(88, 11, 118, 11); //jogging
    break;
}
break;
case 1: //edit
switch (nMenu) {
    case 0:
        u8g.drawFrame(40, 31, 9, 11); // distanced_D_D
        break;
    case 1:
        u8g.drawFrame(48, 31, 9, 11); // distance_D_N
        break;
    case 2:
        u8g.drawFrame(62, 31, 9, 11); // distance_D_F1
        break;
    case 3:
        u8g.drawFrame(28, 43, 9, 11); // distance_U_D
        break;
    case 4:
        u8g.drawFrame(36, 43, 9, 11); // distance_U_N
        break;
    case 5:
        u8g.drawFrame(50, 43, 9, 11); // distance_U_F1
        break;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    break;
}
}
void draw_P1_3(void)
{
    //dip-coating
    u8g.drawRFrame(0, 0, 128, 64, 0);
    u8g.setFont(u8g_font_baby);
    u8g.drawStr(4, 8, "Dip-Coat");
    u8g.drawFrame(40, 0, 124, 14);
    u8g.drawStr(49, 8, "Start");
    u8g.drawFrame(80, 0, 124, 14);
    u8g.drawStr(88, 8, "Jogging");
    /* -----
       แสดงการกด
    ----- */
    if (lisPush)
        u8g.drawStr(10, 25, " ");
    else
        u8g.drawStr(10, 25, " ");
    //menu set
    u8g.setFont(u8g_font_6x12);
    /* -----
       ImmerseTime
    ----- */
    u8g.drawStr(2, 25, "ImmerseTime=");
    sprintf(buffer, "%d", immersetime_T);
    u8g.drawStr (76, 25 , buffer);
    sprintf(buffer, "%d", immersetime_H);
    u8g.drawStr (84, 25, buffer);
    sprintf(buffer, "%d", immersetime_D);
    u8g.drawStr (92, 25 , buffer);
    sprintf(buffer, "%d", immersetime_N);
    u8g.drawStr (100, 25 , buffer);
    u8g.drawStr(110, 25, "s"); // หน่วย

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*-----
    holdTime
    -----*/

    u8g.drawStr(2, 38, "HoldingTime=");
    sprintf(buffer, "%d", holdingtime_T);
    u8g.drawStr (76, 38 , buffer);
    sprintf(buffer, "%d", holdingtime_H);
    u8g.drawStr (84, 38, buffer);
    sprintf(buffer, "%d", holdingtime_D);
    u8g.drawStr (92, 38 , buffer);
    sprintf(buffer, "%d", holdingtime_N);
    u8g.drawStr (100, 38 , buffer);
    u8g.drawStr(110, 38, "s"); // หน่วย
/*-----

    Round
    -----*/

    u8g.drawStr( 3, 52, "Round=");
    sprintf(buffer, "%d", round_D);
    u8g.drawStr (40, 52 , buffer);
    sprintf(buffer, "%d", round_N);
    u8g.drawStr (48, 52 , buffer);
    u8g.drawStr(59, 52, "Cycle");
    u8g.setFont(u8g_font_timR14); //back
    u8g.drawStr(112, 60, "<");
    switch (nMode) {
        case 0://shift
            switch (nMenu) { //เส้นใต้แต่ละตำแหน่ง
                case 0:
                    u8g.drawLine(76, 25, 81, 25); // immersetim_T
                    break;
                case 1:
                    u8g.drawLine(84, 25, 89, 25); // immersetim_H
                    break;
                case 2:
                    u8g.drawLine(92, 25, 97, 25); // immersetim_D
                    break;
            }
        }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 3:
    u8g.drawLine(100, 25, 105, 25); // immersetim_N
    break;
case 4:
    u8g.drawLine(76, 38, 81, 38); // holdindTime_T
    break;
case 5:
    u8g.drawLine(84, 38, 89, 38); // holdindTime_H
    break;
case 6:
    u8g.drawLine(92, 38, 97, 38); // holdindTime_D
    break;
case 7:
    u8g.drawLine(100, 38, 105, 38); // holdindTime_N
    break;
case 8:
    u8g.drawLine(40, 52, 45, 52); // round_D
    break;
case 9:
    u8g.drawLine(48, 52, 53, 52); //round_N
    break;
case 10: //back
    u8g.drawLine(117, 60, 123, 60);
break ;
case 11:
    u8g.drawLine(49, 9, 68, 9); //start
    break;
case 12:
    u8g.drawLine(88, 11, 118, 11); //jogging
    break;
}
break;
case 1: //edit
    switch (nMenu) {
        case 0:
            u8g.drawFrame(74, 16, 9, 11); // immersetim_T

```

```

    break;
case 1:
    u8g.drawFrame(82, 16, 9, 11); // immersetim_H
    break;
case 2:
    u8g.drawFrame(90, 16, 9, 11); // immersetim_D
    break;
case 3:
    u8g.drawFrame(98, 16, 9, 11); // immersetim_N
    break;
case 4:
    u8g.drawFrame(74, 29, 9, 11); // holdindTime_T
    break;
case 5:
    u8g.drawFrame(82, 29, 9, 11); // holdindTime_H
    break;
case 6:
    u8g.drawFrame(90, 29, 9, 11); // holdindTime_D
    break;
case 7:
    u8g.drawFrame(98, 29, 9, 11); // holdindTime_N
    break;
case 8:
    u8g.drawFrame(38, 43, 9, 11); // round_D
    break;
case 9:
    u8g.drawFrame(46, 43, 9, 11); //round_N
    break;
}
break;
}
}
void draw_P2_1(void) {
    //joging
    u8g.drawRFrame(0, 0, 128, 64, 0);
    u8g.drawFrame(0, 0, 40, 14);

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

u8g.setFont(u8g_font_baby);
u8g.drawStr(3, 8, "Dip-Coat");
u8g.drawStr(49, 8, "Start");
u8g.drawFrame(80, 0, 124, 14);
u8g.drawStr(88, 8, "Jogging");

u8g.setFont(u8g_font_helvR08);
if (lsstart == 0)
    u8g.drawStr (44, 40, "START");
else {
    switch (nerr)
    {
    case 0:
        u8g.drawStr (37, 35, "RUNNING");
        u8g.drawStr(33, 51, "Push to stop"); // พิมพ์ข้อความสำหรับหยุดด่วน
        break;
    case -1: // สามารถจำแนก error ได้ถ้ากำหนดขึ้นมาตอนตรวจสอบค่าที่กรอก
        u8g.drawStr(40, 34, "Input error");
        break;
    }
}
/* -----
แสดงการกด
----- */

if (lisPush)
    u8g.drawStr(10, 25, " ");
else
    u8g.drawStr(10, 25, " ");
switch (nMode) {
    case 0://shift
        switch (nMenu) { //เส้นใต้แต่ละตำแหน่ง
            case 0:
                u8g.drawLine(45, 41, 73, 41); // start
                break;
            case 1:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    u8g.drawLine(88, 11, 118, 11); //jogging
    break;
case 2:
    u8g.drawLine(4, 10, 36, 10); //dip
    break;
}
break;
}
}
void draw_P3_1(void) //jogging
{
    // //jogging
    u8g.drawRFrame(0, 0, 128, 64, 0);
    u8g.drawFrame(0, 0, 40, 14); //heater,dip-coa
    u8g.setFont(u8g_font_baby);
    u8g.drawStr(4, 8, "Dip-Coat");
    u8g.drawFrame(0, 0, 80, 14); //dip-coa
    u8g.drawStr(49, 8, "Start");
    u8g.drawStr(88, 8, "Jogging");
    /* -----
       แสดงการกด
    -----
*/
    if (lisPush)
        u8g.drawStr(10, 25, " ");
    else
        u8g.drawStr(10, 25, " ");
    //-----
    --
    // u8g.drawTriangle(72, 36, 64, 18, 56, 36); //motor up
    // u8g.drawTriangle(72, 40, 64, 58, 56, 40); //motor down
    switch (nMode) {
        case 0: //shift
            switch (nMenu) { //เส้นใต้แต่ละตำแหน่ง
                case 0:
                    u8g.drawLine(56, 38, 72, 38); //motor up ok

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    break;
case 1:
    u8g.drawLine(56, 60, 72, 60); //motor down
break;;
case 2:
    u8g.drawLine(49, 9, 68, 9); //start
    break;
case 3:
    u8g.drawLine(4, 10, 36, 10); //dip
    break;
}
break;
case 1: //edit
switch (nMenu) {
case 0:
    u8g.drawFrame(54, 16, 21, 23); //motor up ok
    break;
case 1:
    u8g.drawFrame(54, 38, 21, 23); //motor down
    break;
}
break;
}
}

```





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

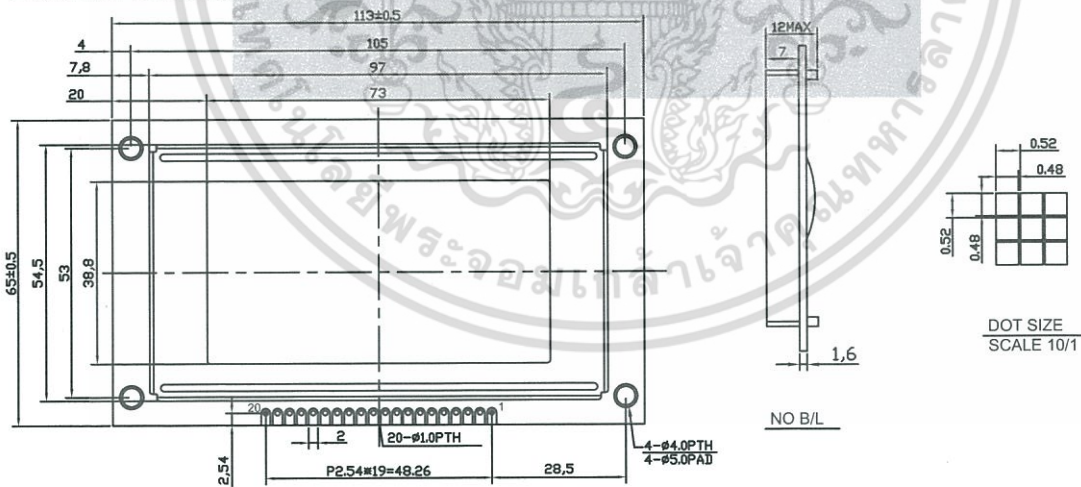
## Chapter1. 12864 Graphic LCD Module Brief

### Introduction

#### Physical Data

Item	Contents	Unit
LCD type	STN	---
LCD duty	1/64	---
LCD bias	1/9	---
Viewing direction	6	o'clock
Module size (W×H×T)	112×66	mm
Viewing area (W×H)	73.4 × 38.8 (2.89" × 1.53")	mm
Number of dots	128 × 64	dots
Dot size (W×H)	0.48 × 0.48 (0.019" × 0.019")	mm
Dot pitch (W×H)	0.52 × 0.52 (0.020" × 0.020")	mm

#### External Dimensions



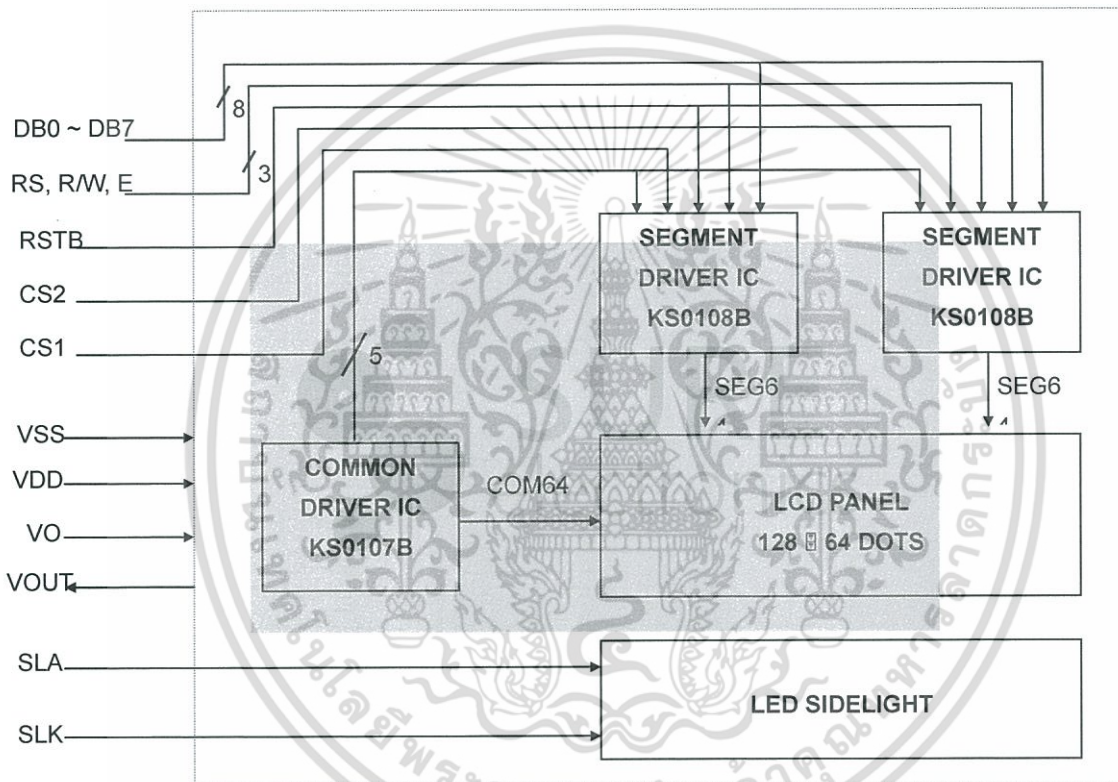
#### Mechanical Data

Item	Standard Value	Unit
Module Dimension	113×65	mm
Viewing Area	73×38.8	mm
Mounting hole	105×53	mm
Dot pitch	0.52×0.52	mm

## 12864 Graphic LCD Module Brief Introduction

### Block Diagram

20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
SL	SL	VO	RS	CS	CS	DB	DB	DB	DB	DB	DB	DB	DB	E	R/W	RS	VO	VD	VS
K	A	UT	TB	2	1	7	6	5	4	3	2	1	0					D	S



### Absolute Maximum Ratings (Ta = 25°C)

Parameter	Symbol	Min	Max	Unit
Supply voltage for logic	VDD	-0.3	7.0	V
Supply voltage for LCD	VDD - VO	-0.3	VDD+0.3	V
Input voltage	VI	-0.3	VDD+0.3	V
Operating temperature	TOP	0	50	°C
Storage temperature	TST	-10	60	°C

## 12864 Graphic LCD Module Brief Introduction

### Electrical Characteristics (VDD = +5V±5%, VSS = 0V, Ta = 25°C)

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Supply voltage for logic	VDD	---	4.5	5.0	5.5	V
Supply current for logic	IDD	---	---	2.02	4	mA
Operating voltage for LCD	VDD - VO	0°C	8.4	8.8	9.2	V
		25°C	8.1	8.5	8.9	V
		50°C	7.4	7.8	8.2	V
Supply voltage for sidelight	VF	---	---	4.2	4.6	V
Supply current for sidelight	IF	VF=4.2V	---	180	300	mA
Input voltage 'H' level	VIH	---	0.7VDD	---	VDD	V
Input voltage 'L' level	VIL	---	0	---	0.3VDD	V

### Interface Pin Connections

Pin No.	Symbol	Level	Description
1	VSS	0V	Ground
2	VDD	5.0V	Supply voltage for logic
3	VO	---	Input voltage for LCD
4	RS	H/L	H : Data signal, L : Instruction signal
5	R/W	H/L	H : Read mode, L : Write mode
6	E	H, H → L	Chip enable signal
7	DB0	H/L	Data bit 0
8	DB1	H/L	Data bit 1
9	DB2	H/L	Data bit 2
10	DB3	H/L	Data bit 3
11	DB4	H/L	Data bit 4
12	DB5	H/L	Data bit 5
13	DB6	H/L	Data bit 6
14	DB7	H/L	Data bit 7
15	CS1	H	Chip select signal for KS0108B(1)
16	CS2	H	Chip select signal for KS0108B(2)
17	RSTB	L	Reset signal
18	VOUT	-5V	Output voltage for LCD
19	SLA	4.2V	Sidelight anode
20	SLK	0V	Sidelight cathode

---

## Chapter2. Contact Us

---

**Sure Electronics Co., Ltd.**

5F, Zone A,

Qinhuai Technology Innovation Center

105-2 DaMing Rd (Zip Code: 210022)

Nanjing

PRC

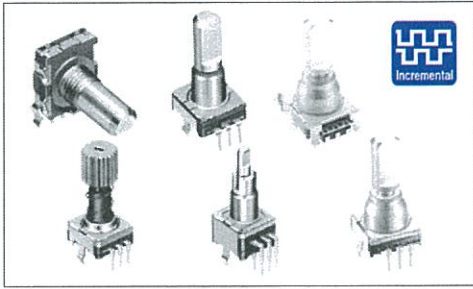
Tel: +86-25-66606340 (English Service) GMT1am-10am

Fax: +86-25-66606346

Website: [www.sure-electronics.net](http://www.sure-electronics.net)[www.sureelectronics.net](http://www.sureelectronics.net)[www.sure-electronics.com](http://www.sure-electronics.com)



Compact and highly reliable type available in many varieties.



Typical Specifications

Items	Specifications
Output signal	Two phase of A, B Three phase of A, B and C (EC11EH) Self-return switch (EC111 / EC11E0B)
Rating	10mA 5V DC
Operating life	15,000 cycles 30,000 cycles (EC11EH) 100,000 cycles (EC11K / EC11J)
Operating temperature range	- 40°C to + 85°C

Detector

Slide

Push

Rotary

Encoders

Power

Dual-in-line  
Package Type

TACT Switch™

Metal  
Shaft

Insulated  
Shaft

Hollow  
Shaft

Ring  
Type

Product Line

Structure	Shaft configuration	Length of the shaft (L <sub>1</sub> ) (mm)	Torque (mN·m)	Number of detent	Number of pulse	Push-on switch	Travel of push-on switch (mm)	Operating life (cycles)	Minimum order unit (pcs.)		Product No.	Drawing No.							
									Japan	Export									
Horizontal			12 ± 7	30	15	Without	—		700	1,400	EC11B15202AA	1							
						With	0.5				EC11B15242AE	2							
							1.5				EC11B15242AF	3							
Vertical	Flat	20	10 ± 7	18	9			15,000	1,200	2,400	EC11E09204A4	4							
				30	15						EC11E15204A3								
				Without	Without						EC11E1530401								
			10 ± 7	36	18						EC11E1820402								
			7 ± 3/4	Without	18						EC11E1830401								
			10 ± 7	18	9						EC11E09244BS								
			7 ± 3/4	Without	15						EC11E15244G1								
			10 ± 7	36	18						EC11E153440D								
			7 ± 3/4	Without	18						EC11E18244AU								
			10 ± 7	30	15						EC11E183440C								
			7 ± 3/4	Without	9						EC11E09244AQ								
			10 ± 7	36	18						EC11E15244B2								
			Less shaft wobble	Serrated	25	12 ± 7	30				15		Without	—		1,000	2,000	EC11E1534408	5
							With						1.5	EC11E18244A5					
														EC11E1834403					
Vertical	Flat	20	12 ± 5	18	9			100,000	1,000	2,000	EC11G1560414	6							
				30	15						Without		—	EC11G1574402					
				18	9						With		0.5	EC11G1564411					
				30	15	EC11K0920401													
				18	9	EC11K1520401													
				30	15	1.5					EC11K0924401								
18	9	EC11K1524402																	
30	15	EC11K0925401																	
											EC11K1525401	10							

Note

Other varieties are also available. Please inquire. Refer to P.167 for product specifications. Refer to P.167 for attached parts.

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้เท่านั้น Refer to P.168 for product varieties. Refer to P.193 for soldering conditions.

Product Line

- Detector
- Slide
- Push
- Rotary
- Encoders
- Power
- Dual-in-line Package Type
- TACT Switch™
- Metal Shaft
- Insulated Shaft
- Hollow Shaft
- Ring Type

Structure	Shaft configuration	Length of the shaft (L <sub>1</sub> , L <sub>2</sub> ) (mm)	Torque (mN·m)	Number of detent	Number of pulse	Push-on switch	Travel of push-on switch (mm)	Operating life (cycles)	Minimum order unit (pcs.)		Product No.	Drawing No.	
									Japan	Export			
Reflow	Flat	20	12 ± 5	18	9	Without	—	100,000	300	600	EC11J0920404	11	
				30	15						EC11J152040F		
				18	9	With	0.5				EC11J0924411	12	
				30	15						EC11J1524413		
				18	9						EC11J0925403		
											EC11J1525402		
Push lock	20-tooth Serrated	25	10 ± 7	30	15	Without	—	15,000	800	1,600	EC11E152T409	13	
		26.4			With	8	EC11E152U402				14		
Self-return switch	Flat	15	3 to 30	Without	Self-return switch	Without	—	30,000	1,200	2,400	EC1110120005	15	
							0.5				EC111012010H		16
							1.5				EC1110120201		
ABC switch	Flat	20	10 ± 7	18	ABC switch	With	0.5	30,000	1,200	2,400	EC11EH124403	17	
							30				EC11EH224403		
							18				EC11EH124404		
		1.5	EC11EH224404										
Dual-shaft	Inner-shaft=25	30	30	15	Without	Without	0.5	15,000	700	1,400	EC11EBB24C03	18	
	Slotted Outer-shaft=15						Without				—		
	Flat Inner-shaft=25						With				1.5		EC11E0B2LB01
Slotted Outer-shaft=15	3 to 30	Without	Self-return switch	Without	—								

Note

Other varieties are also available. Please inquire.

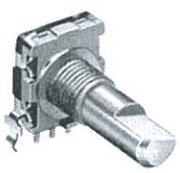
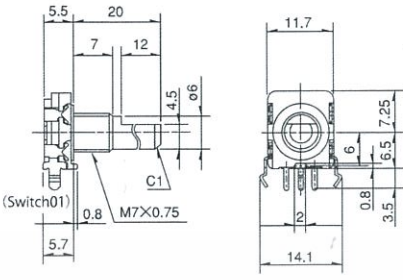
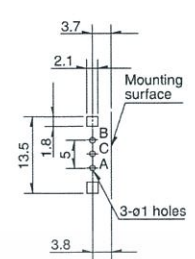
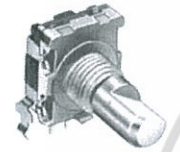
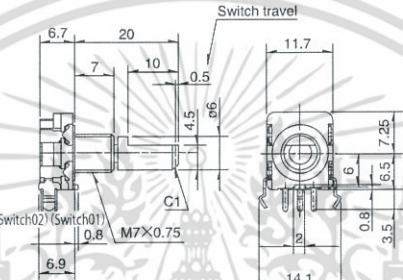
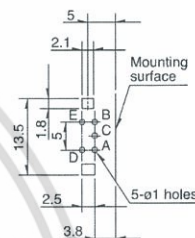

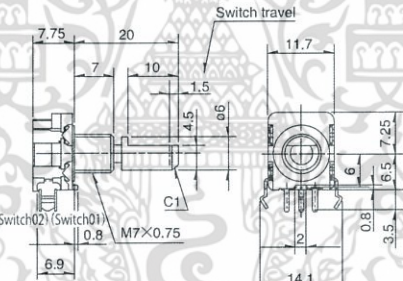
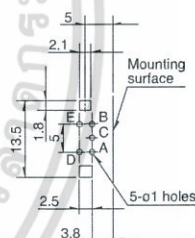
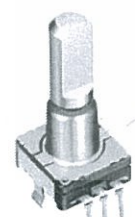
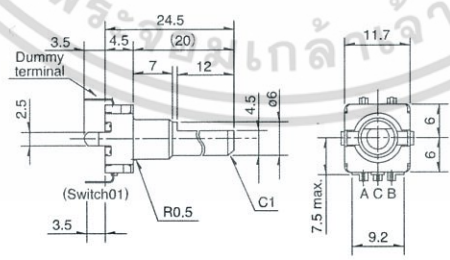
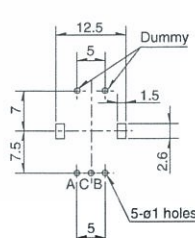
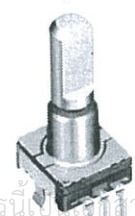
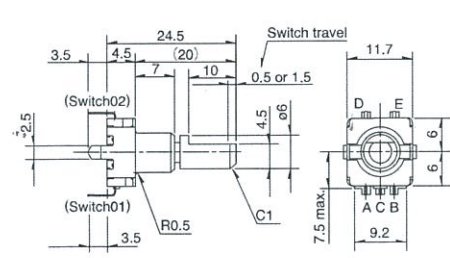
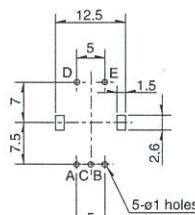
Packing Specifications Tray

Product No.	Number of packages (pcs.)		Export package measurements (mm)
	1 case / Japan	1 case / export packing	
EC11B	700	1,400	370 × 520 × 201
EC11E09 / 15 / 18 / EC11EH	1,200	2,400	540 × 360 × 250
EC11G / EC11K	1,000	2,000	
EC11J	300	600	369 × 283 × 263
EC11E152T / U	800	1,600	363 × 507 × 230
EC111	1,200	2,400	363 × 507 × 216
EC11E0B / BB	700	1,400	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dimensions

Unit:mm

No.	Photo	Style	PC board mounting hole dimensions (Viewed from mounting side)
1	 <p>EC11B Horizontal</p>	 <p>5.5 20 7 12 4.5 0.6 11.7 6 6.5 7.25 0.8 M7X0.75 5.7 14.1 2 0.8 3.5</p>	 <p>3.7 2.1 Mounting surface 13.5 1.8 5 B C A 3-ø1 holes 3.8</p>
2	 <p>EC11B Horizontal with push-on switch (travel 0.5mm)</p>	 <p>6.7 20 7 10 0.5 11.7 6 6.5 7.25 0.8 M7X0.75 6.9 14.1 2 0.8 3.5 Switch travel C1</p>	 <p>5 2.1 Mounting surface 13.5 1.8 5 E B C A D 5-ø1 holes 2.5 3.8</p>
3	 <p>EC11B Horizontal with push-on switch (travel 1.5mm)</p>	 <p>7.75 20 7 10 1.5 11.7 6 6.5 7.25 0.8 M7X0.75 6.9 14.1 2 0.8 3.5 Switch travel C1</p>	 <p>5 2.1 Mounting surface 13.5 1.8 5 E B C A D 5-ø1 holes 2.5 3.8</p>
4	 <p>EC11E Vertical</p>	 <p>3.5 4.5 24.5 (20) 7 12 4.5 0.6 11.7 6 6 7.5 max. 9.2 3.5 2.5 Dummy terminal (Switch01) R0.5 C1 A C B</p>	 <p>12.5 Dummy 5 1.5 7.5 7 2.6 5-ø1 holes 5 A C B</p>
5	 <p>EC11E Vertical with push-on switch (travel 0.5 / 1.5mm)</p>	 <p>3.5 4.5 24.5 (20) 7 10 0.5 or 1.5 11.7 6 6 7.5 max. 9.2 3.5 2.5 (Switch02) (Switch01) R0.5 C1 D E A C B</p>	 <p>12.5 5 1.5 7.5 7 2.6 5-ø1 holes 5 D E A C B</p>

Detector

Slide

Push

Rotary

Encoders

Power

Dual-in-line  
Package Type

TACT Switch™

Metal  
Shaft

Insulated  
Shaft

Hollow  
Shaft

Ring  
Type

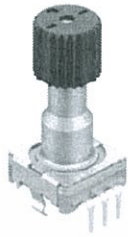
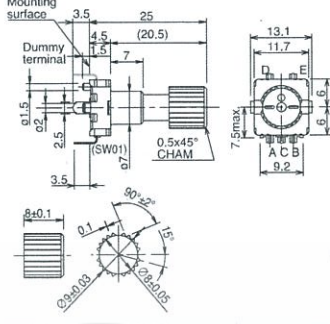
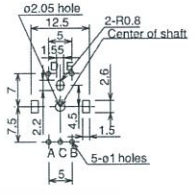
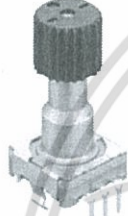
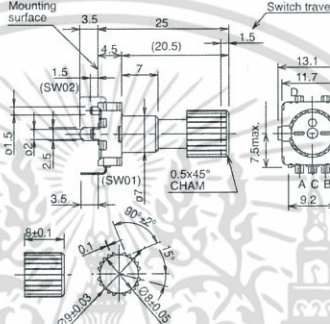
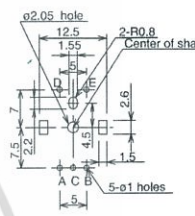
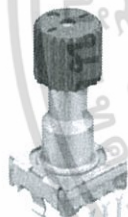
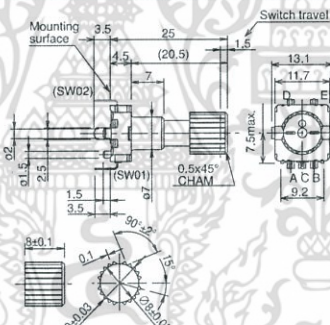
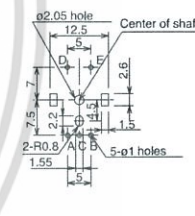

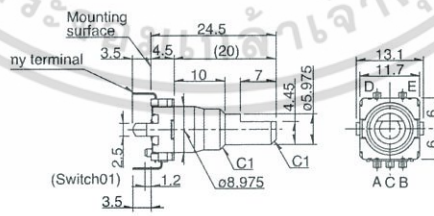
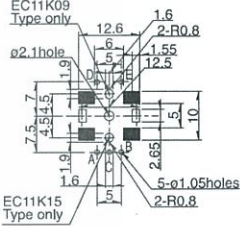

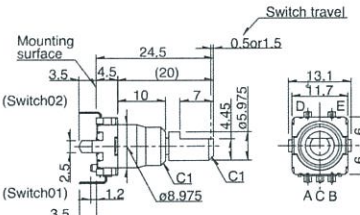
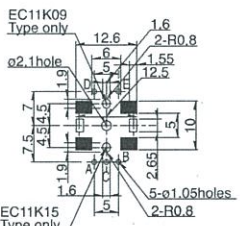
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น ออกทั้งหมดมีเหตุเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dimensions

Unit:mm

- Detector
- Slide
- Push
- Rotary
- Encoders
- Power
- Dual-in-line Package Type
- TACT Switch™
- Metal Shaft
- Insulated Shaft
- Hollow Shaft
- Ring Type

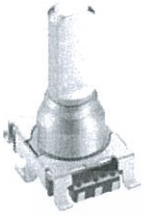
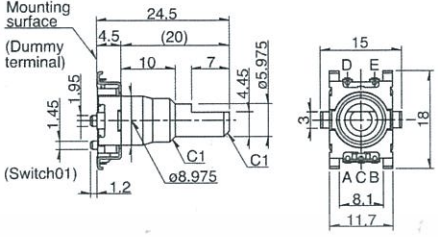
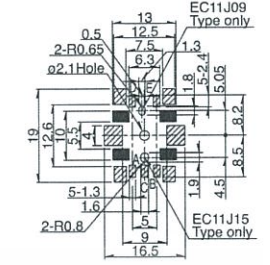

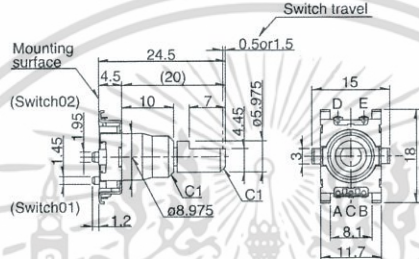
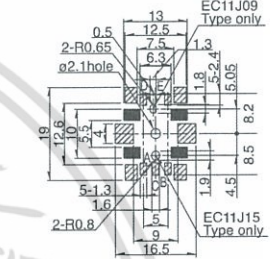

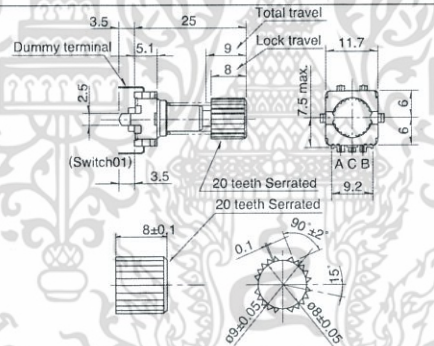
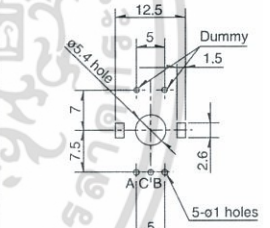

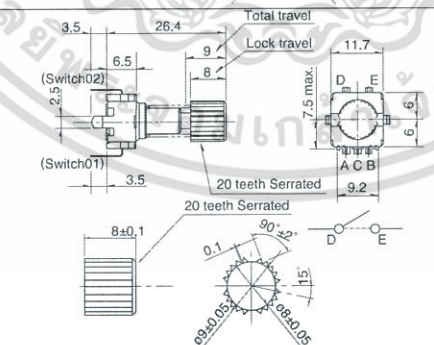
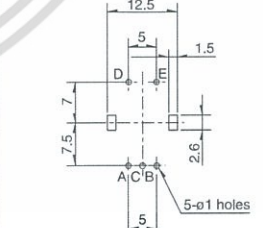
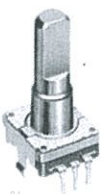
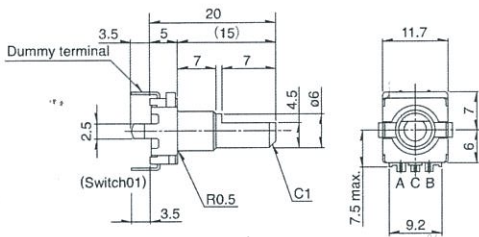
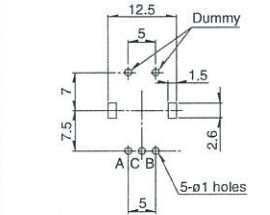
No.	Photo	Style	PC board mounting hole dimensions (Viewed from mounting side)
6			
7			
8			
9			 <p>Black part : Do not solder and no wiring for electrical contact.</p>
10			 <p>Black part : Do not solder and no wiring for electrical contact.</p>

เอกสารนี้เป็นเอกสารของบริษัทผู้ผลิตไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น ออกทั้งหมดมเหตุดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรนำไปใช้

Dimensions

Unit:mm

No.	Photo	Style	PC board mounting hole dimensions (Viewed from mounting side)
11	 <p><b>EC11J Reflow</b></p>	 <p>Mounting surface (Dummy terminal) (Switch01) 24.5 (20) 4.5 10 7 4.45 0.5.975 1.45 1.95 1.2 08.975 C1 C1 15 18 3 D E F A C B 8.1 11.7</p>	 <p>EC11J09 Type only EC11J15 Type only 13 12.5 7.5 6.3 1.3 1.8 5.2.4 5.05 1.9 8.2 8.5 16.5 5-1.3 1.6 2-R0.8 19 12.6 10 4 2-R0.65 ø2.1Hole A C B 5 9 16.5</p> <p>A slant line part: The solder land Black part: Do not solder and wiring for electrical contact</p>
12	 <p><b>EC11J with push-on switch (travel 0.5mm / 1.5mm)</b></p>	 <p>Switch travel Mounting surface (Switch02) (Switch01) 24.5 (20) 4.5 10 7 4.45 0.5.975 1.45 1.95 1.2 08.975 C1 C1 15 18 3 D E F A C B 8.1 11.7 0.5or1.5</p>	 <p>EC11J09 Type only EC11J15 Type only 13 12.5 7.5 6.3 1.3 1.8 5.2.4 5.05 1.9 8.2 8.5 16.5 5-1.3 1.6 2-R0.8 19 12.6 10 4 2-R0.65 ø2.1Hole A C B 5 9 16.5</p> <p>A slant line part: The solder land Black part: Do not solder and wiring for electrical contact</p>
13	 <p><b>EC11E152T Push-lock mechanism</b></p>	 <p>Dummy terminal (Switch01) Total travel Lock travel 20 teeth Serrated 20 teeth Serrated 8±0.1 0.1 90±2 08±0.05 08±0.05 3.5 5.1 25 9 11.7 7.5 max. 9.2 6.6 A C B 5-1 holes</p>	 <p>12.5 5 1.5 Dummy ø5.4 hole 7.5 7 2.6 5 A C B 5-ø1 holes</p>
14	 <p><b>EC11E152U Push-lock mechanism with push-lock switch</b></p>	 <p>(Switch02) (Switch01) Total travel Lock travel 20 teeth Serrated 20 teeth Serrated 8±0.1 0.1 90±2 08±0.05 08±0.05 3.5 6.5 26.4 9 11.7 7.5 max. 9.2 6.6 D E F A C B 5-1 holes</p>	 <p>12.5 5 1.5 7.5 7 2.6 5 A C B 5-ø1 holes</p>
15	 <p><b>EC111 Self-return switch</b></p>	 <p>Dummy terminal (Switch01) 20 (15) 7 7 4.5 ø6 R0.5 C1 11.7 7.5 max. 9.2 6 7 A C B</p>	 <p>12.5 5 1.5 7.5 7 2.6 5 A C B 5-ø1 holes Dummy</p>

Detector

Slide

Push

Rotary

Encoders

Power

Dual-in-line Package Type

TACT Switch™

Metal Shaft

Insulated Shaft

Hollow Shaft

Ring Type

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

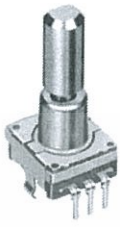
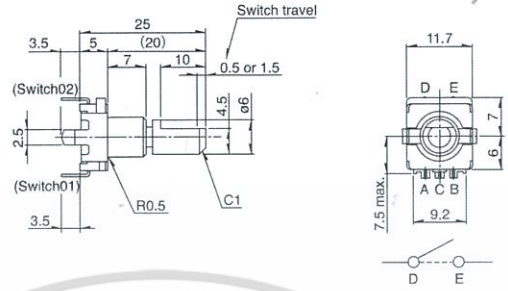
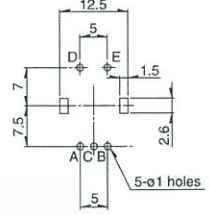

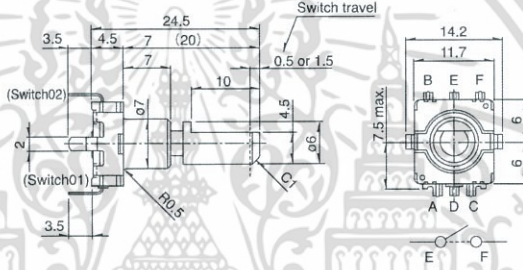
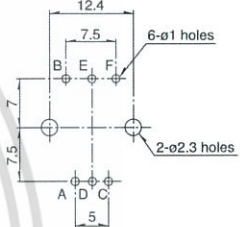
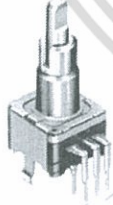
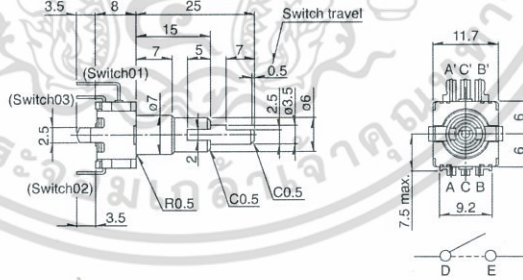
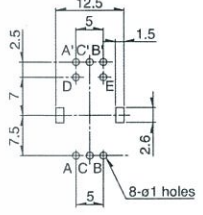
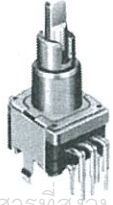
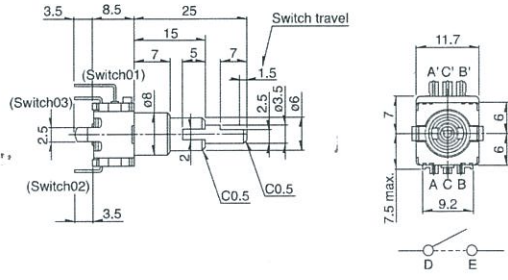
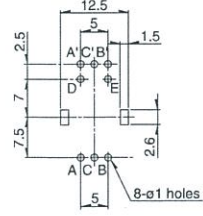
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dimensions

Unit:mm

- Detector
- Slide
- Push
- Rotary
- Encoders
- Power
- Dual-in-line Package Type
- TACT Switch™

- Metal Shaft
- Insulated Shaft
- Hollow Shaft
- Ring Type

No.	Photo	Style	PC board mounting hole dimensions (Viewed from mounting side)
16	<p><b>EC111</b> Self-return switch with push-on switch (travel 0.5mm / 1.5mm)</p> 		
17	<p><b>EC11EH</b> ABC switch with push-on switch (travel 0.5mm / 1.5mm)</p> 		
18	<p><b>EC11E</b> Dual-shaft type with push-on switch (travel 0.5mm) Inner shaft: encoder Outer shaft: encoder</p> 		
19	<p><b>EC11E</b> Dual-shaft type with push-on switch (travel 1.5mm) Inner shaft: encoder Outer shaft: self-return switch</p> 		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งหากมีการนำไปใช้

# 11mm Size Metal Shaft Encoders / Product Specifications

## Output Wave

EC11B / EC11E / EC11G	EC11E / EC11J / EC11K	EC11EH
<p>EC11B, EC11E, EC11G 30 detents, 15 pulse</p> <p>Detent stability position CW direction →</p>	<p>Detent stability position cannot be specified for B signal.</p> <p>EC11E 18 detents 9 pulse EC11E 36 detents 18 pulse EC11J, EC11K</p> <p>Detent stability position CW direction →</p>	<p>Detent position CW direction →</p>

Detector

Slide

Push

Rotary

Encoders

Power

Dual-in-line  
Package Type

TACT Switch™

## Sliding Noise

Test circuit		Output waveform		Detailed dimensions	
<p>Measurement condition : Rotation speed 360°/s t : Masking time to avoid chattering</p>		<p>EC11B</p> <p><math>V_1=V_2=1.5V</math> max. At <math>R = 5k\Omega</math> Chattering : 2ms max. Bounce : 2ms max.</p>	<p>EC11E / EC11G</p> <p><math>V_1=V_2=1.5V</math> max. At <math>R = 5k\Omega</math> Chattering : 3ms max. Bounce : 2ms max.</p>	<p>EC11J / EC11K</p> <p><math>V_1=V_2=2.5V</math> max. At <math>R = 5k\Omega</math> Chattering : 3ms max. Bounce : 2ms max.</p>	
<p>Test condition : Rotation speed 360°/s</p>		<p>EC11EH</p> <p>At <math>R = 5k\Omega</math> Chattering : 5ms max. Bounce : 5ms max.</p>			

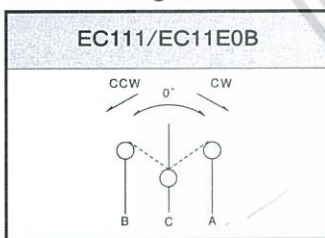
Metal  
Shaft

Insulated  
Shaft

Hollow  
Shaft

Ring  
Type

## Circuit Diagram

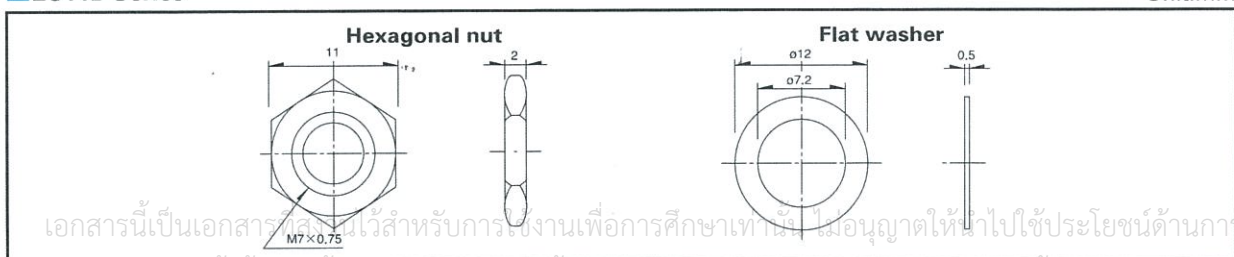


# 11mm Size Metal Shaft Encoders / Attached Parts

The following parts are included with the product.

## EC11B Series

Unit:mm



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Shaft Dimensions**

**1. Single-shaft Type (EC11K is excluded. Refer to P.212 for EC11K)**

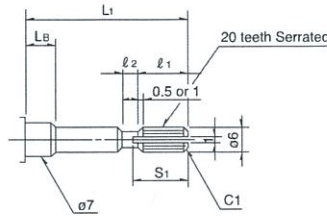
**1) Serrated Type**

Unit:mm

**EC11**

**Configuration (Shaft diameter :  $\phi$  6)**

Not applicable for EC11E and EC11G with push-lock mechanism

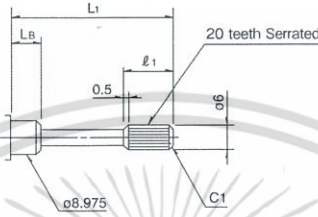


**Detailed dimensions**

L <sub>1</sub>	L <sub>B</sub>	l <sub>1</sub>	l <sub>2</sub>	S <sub>1</sub>
20	7	6	1	7
25	10	10	2	11

**EC11K / EC11J**

**Configuration (Shaft diameter :  $\phi$  6)**



**Detailed dimensions**

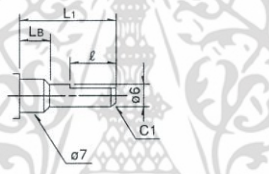
L <sub>1</sub>	L <sub>B</sub>	l <sub>1</sub>
15	7	5
20	10	6

**2) Flat Type**

Unit:mm

**EC11**

**Configuration (Shaft diameter :  $\phi$  6)**



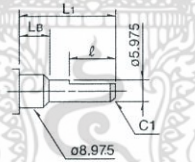
**Detailed dimensions**

L <sub>1</sub>	L <sub>B</sub>	l
※1 15	5	7
15	7	5 (6)
20	7	10 (12)
※2 25	10	12

※ 1 Does not comply with EC111  
 ※ 2 L<sub>B</sub>=7 for EC11B.  
 Values in parentheses apply to products without push-on switch.

**EC11K / EC11J**

**Configuration (Shaft diameter :  $\phi$  5.975)**



**Detailed dimensions**

L <sub>1</sub>	L <sub>B</sub>	l
15	7	5
20	10	7

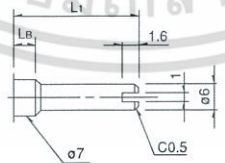
Dimensions marked with ( ) apply to products without push-on switches.

**3) Slotted Type**

Unit:mm

**EC11**

**Configuration (Shaft diameter :  $\phi$  6)**



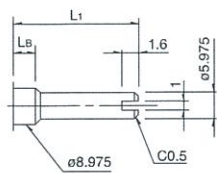
**Detailed dimensions**

L <sub>1</sub>	L <sub>B</sub>
15	7
20	7
※ 25	10

※ L<sub>B</sub>=7 for EC11B.

**EC11K / EC11J**

**Configuration (Shaft diameter :  $\phi$  5.975)**



**Detailed dimensions**

L <sub>1</sub>	L <sub>B</sub>
15	7
20	10

**Notes**

- The highlighted figures in shaft types refer to Product Specifications in P.161 and P.162.
- Other varieties are also available. Please inquire.

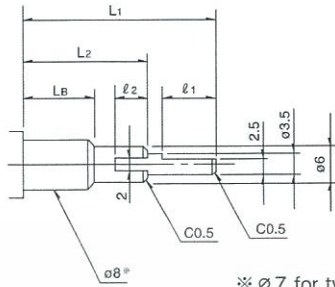
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Standard Dimensions and configuration of Dual-shaft Type

1) Flat Type

Unit:mm

Configuration (Inner-shaft :  $\phi$  3.5 Outer-shaft :  $\phi$  6)



Detailed dimensions

L <sub>1</sub>	L <sub>2</sub>	L <sub>B</sub>	l <sub>1</sub>	l <sub>2</sub>
20	10	5	7	4
25	15	7	7	5
30	20	10	7	5

※  $\phi$  7 for type with inner / outer shaft encoders (EC11EBB)

Notes

- The highlighted figures in shaft types refer to Product Specifications in P.162
- Other varieties are also available. Please inquire.



Detector

Slide

Push

Rotary

Encoders

Power

Dual-in-line  
Package Type

TACT Switch™

Metal  
Shaft

Insulated  
Shaft










Hollow  
Shaft

Ring  
Type

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# Encoders

## List of Varieties

Type		Metal shaft															
		9mm size			11mm size												
Series		EC09E		EC11B		EC11E			EC11G								
Photo																	
Output		Incremental (Two phase A and B)															
Shaft types		Single-shaft						Dual-shaft		Single-shaft							
Operating direction		Vertical		Horizontal		Vertical											
Number of pulse / Number of detent		15 / 30			9 / 18 15 / 30 or Without 18 / 36 or Without			15 / 30 or Without									
Features		—		—		Without detent Push-lock mechanism		—		Less shaft wobble							
Dimensions (mm)		W		9.5		11.7											
		D		—		13.8		12			—						
		H		4.5		5.5		4.5		8 / 8.5	4.5						
Operating temperature range		-40°C to + 85°C															
Operating life		15,000 cycle															
Automotive use		●		●		●			●								
Life cycle (availability)																	
Electrical performance		Rating		10mA 5V DC													
		Max./min. operating current (Resistive load)		10mA / 1mA													
		Insulation resistance		100M Ω min. 250V DC													
		Voltage proof		300V AC for 1 minute or 360V AC for 1s		300V AC for 1 minute or 360V AC for 2s											
Mechanical performance		Rotational torque (Without detent)		—		7 ± <sub>4</sub> mN·m			8.5 ± 5 mN·m								
		Detent torque		8 ± 5mN·m		12 ± 7mN·m		10 ± 7mN·m			12 ± 7 mN·m						
		Push-pull strength		100N													
Shaft configuration		Flat		Flat, Slotted, Serrated			Inner-shaft: Flat Outer-shaft: Slotted		Serrated								
Terminal type		Insertion															
Switch Specifications		Switch type		Push-on switch			Push-lock mechanism switch※		Push-on switch								
		Contact arrangement		Single pole and single throw (Push-on)													
		Travel (mm)		0.5 ± 0.3		1.5 ± 0.5		0.5 ± <sub>0.3</sub> <sup>+0.4</sup>		1.5 ± 0.5		0.5 ± 0.3	1.5 ± 0.5	1.5 ± 0.35			
		Operating force (N)		6 ± <sub>2</sub> <sup>+2.5</sup>		4 ± 2		6 ± 3		5 ± 2		6 ± <sub>2</sub> <sup>+2.5</sup>	4 ± 2	8 max.	6 ± <sub>2</sub> <sup>+2.5</sup>	4 ± 2	5 ± 2
		Rating		10mA 5V DC		0.1A 5V DC (500 μA 5V DC min. ratings)											
		Contact resistance		100m Ω max. for initial period; 200m Ω max. after operating life.													
		Operating life		10,000 times min.		25,000 times min.		20,000 times min.		10,000 times min.	20,000 times min.						
Page		160			161												

- Encoders Soldering Conditions . . . . . 193
- Encoders Cautions . . . . . 194

### Notes

- ※ marked specification is only applicable to EC11E152U402.
- indicates applicability to all products in the series.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Detector  
Slide  
Push  
Rotary  
Encoders  
Power  
Dual-in-line  
Package Type  
TACT Switch™

Metal Shaft  
Insulated Shaft  
Hollow Shaft  
Ring Type

# Encoders

## List of Varieties

Type	Metal shaft						
	11mm size				20mm size		
Series	EC111	EC11EH	EC11K	EC11J	EC20A / RK203	EM20B	
Detector	Photo						
Slide	Photo						
Push	Output						
Rotary	Shaft types						
Encoders	Operating direction						
Power	Number of pulse / Number of detent						
Dual-in-line Package Type	Features						
TACT Switch™	Dimensions (mm)						
	W	11.7				20.2	20
	D	13	12		14.2	19.2	22.25
	H	5	4.5			10	13
	Operating temperature range						
	Operating life						
	Automotive use						
	Life cycle (availability)						
Metal Shaft	Electrical performance						
	Rating						
	Max./min. operating current (Resistive load)						
	Insulation resistance						
Insulated Shaft	Voltage proof						
	Rotational torque (Without detent)						
	Detent torque						
Hollow Shaft	Push-pull strength						
	Shaft configuration						
Ring Type	Terminal type						
	Switch type						
	Contact arrangement						
	Travel (mm)						
	Operating force (N)						
	Rating						
	Contact resistance						
	Operating life						
	Page						

- Encoders Soldering Conditions . . . . . 193
- Encoders Cautions . . . . . 194

**Note** ● indicates applicability to all products in the series.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Encoders Soldering Conditions

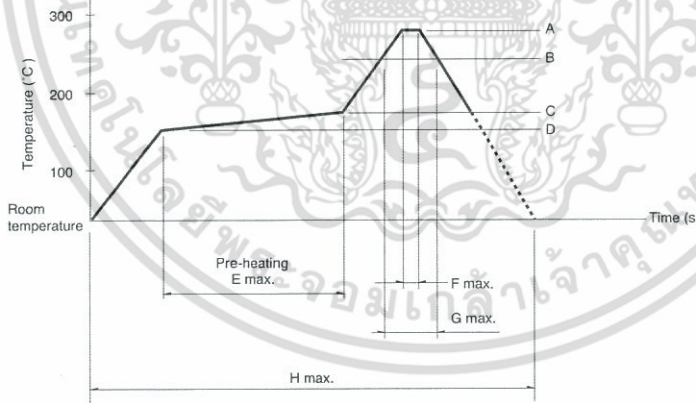
### Reference for Hand Soldering

Series	Tip temperature	Soldering time	No. of solders
EC05E, EC09E, EC10E, EC111, EC11B, EC11E, EC11G, EC11K, EC12D, EC12E, EC18A, EC20A, EC21A, EC28A, EC35A, EC35AH, EC35B, EC40A, EC45A, EC50A, EC60A, EM20B	350°C max.	3s max.	1 time
EC11J	350±10°C	3 <sup>+1</sup> <sub>0</sub> s	2 time

### Reference for Dip Soldering

Series	Preheating		Dip soldering		No. of solders
	Soldering surfacetemperature	Heating time	Soldering temperature	Soldering time	
EC09E, EC11B, EC111, EC11E, EC11G, EC11K, EC18A, EC20A, EC21A, EC28A, EC35A, EC35AH, EC35B, EC50A	100°C max.	2 min. max.	260±5°C	5±1s	2 time max.
EC10E, EC12D, EC12E	100°C max.	1 min. max.	260±5°C	3±1s	2 time max.
EC40A	110°C max.	1 min. max.	260°C max.	10s max.	1 time
EC45A, EC60A	100°C max.	2 min. max.	260°C max.	5s max.	2 time max.
EM20B	80°C max.	1 min. max.	260°C max.	3s max.	2 time max.

### Condition for Reflow Temperature profile



Series	A	B	C	D	E	F	G	H	No. of reflows
EC11J	260°C	230°C	180°C	150°C	2 min.max.	3s	40s	4 min. max	2 time max.
EC05E	250°C min.	230°C min.	180°C	150°C	60s to 120s	—	30s to 40s	—	2 time max.

### Notes

- When using an infrared reflow oven, solder may sometimes not be applied. Be sure to use a hot air reflow oven or a type that uses infrared rays in combination with hot air.
- The temperatures given above are the maximum temperatures at the terminals of the potentiometer when employing a hot air reflow method. The temperature of the PC board and the surface temperature of the potentiometer may vary greatly depending on the PC board material, its size and thickness. Ensure that the surface temperature of the potentiometer does not rise to 250°C or greater.
- Conditions vary to some extent depending on the type of reflow bath used. Be sure to give due consideration to this prior to use.

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Detector

Slide

Push

Rotary

Encoders

Power

Dual-in-line  
Package Type

TACT Switch™

Metal  
Shaft

Insulated  
Shaft

Hollow  
Shaft

Ring  
Type

### 3. Names and Functions of Driver Parts

#### Driver Front Panel

Standard Type Driver: UDK5114N

High Speed Type Driver: UDK5128N

Electromagnetic Brake

Standard Type Driver: UDK5114N-M

Electromagnetic Brake

High Speed Type Driver: UDK5128N-M

Standard Type Driver: UDK5107N

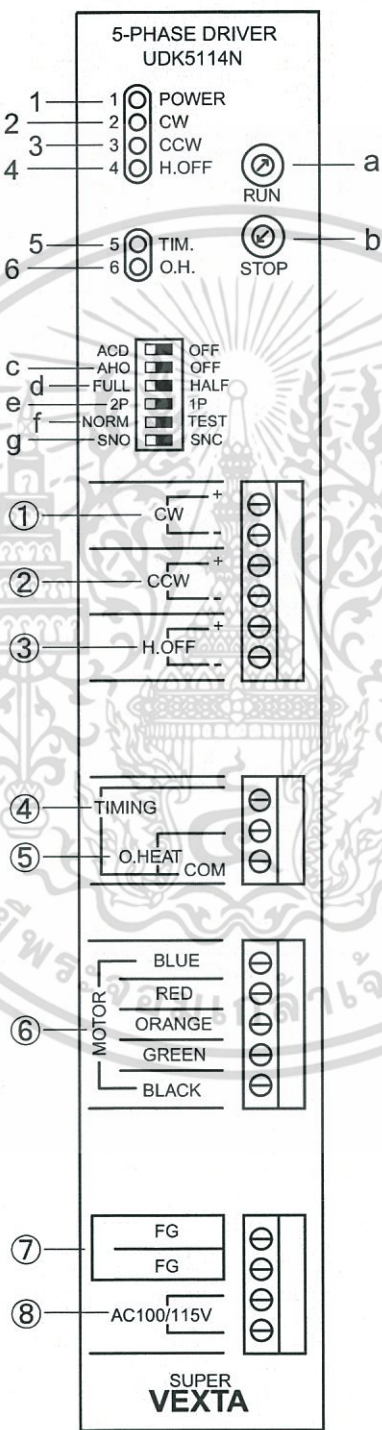
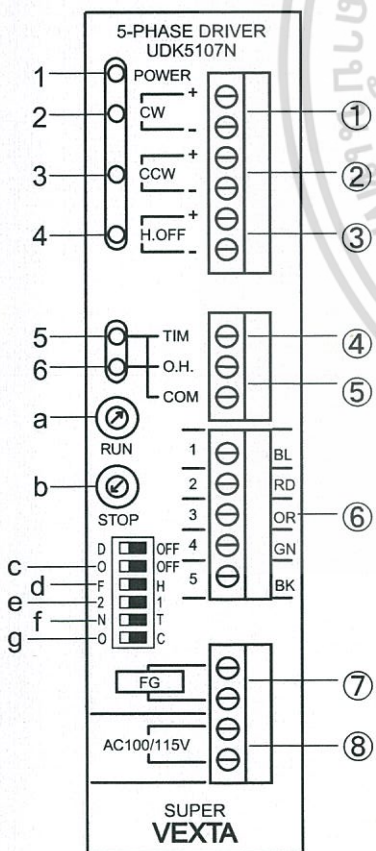


Illustration shows UDK5114N.

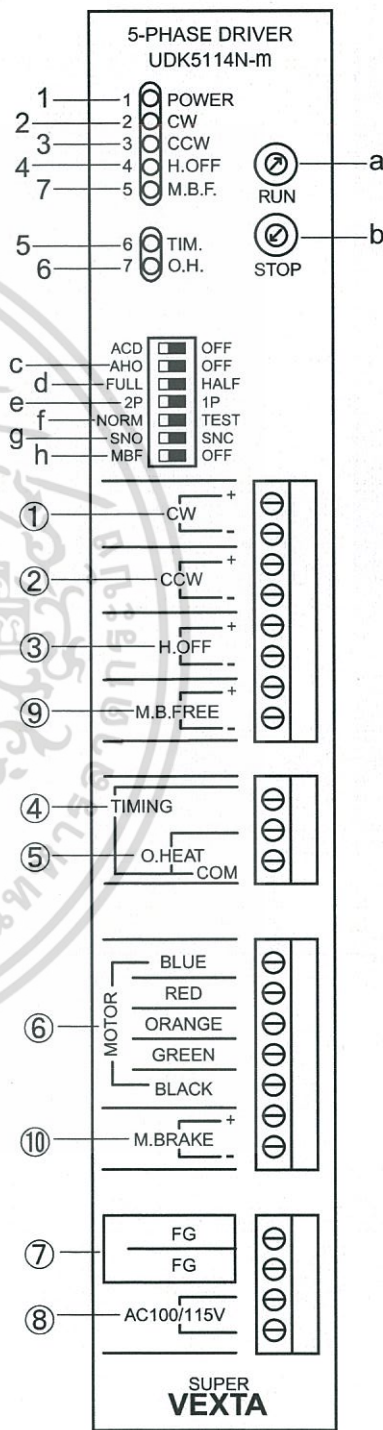


Illustration shows UDK5114N-M.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1 LED Indicators

The LED indicators show the state of various input/output signals etc.

They are indicated on page 6 as 1 ~ 6 (1 ~ 7 for the electromagnetic brake type).

Information within the brackets [ ] refers to driver model UDK5107N only.

LED Name	Indication	Color	Condition When LED ON	Page Reference
1 Power Input LED	POWER	Green	Lights when single phase 100V ± 15% (50/60Hz) or single phase 115V ± 15% (60Hz) is input.	—
2 CW Pulse Signal Input LED	CW	Green	Lights when a CW pulse signal is input. (In 1 pulse input mode, indicates a pulse signal is input.)	Page 22, 23
3 CCW Pulse Signal Input LED	CW	Green	Lights when a CCW pulse signal is input. (In 1 pulse input mode, indicates a rotation direction signal is input)	Page 22, 23
4 Output Current Off Signal Input LED	H.OFF	Green	Lights when the output current off signal is input.	Page 24
5 Excitation Timing Signal Output LED	TIM.	Green	Lights when the excitation timing signal is output.	—
6 Overheat Signal Output LED	O.H.	Red	Lights when the overheat signal is output.	Page 27, 28
7 Electromagnetic Brake Release Signal Input LED (For electromagnetic brake type only)	M.B.F.	Green	Lights when the electromagnetic brake release signal is input.	Page 25, 26

### 3.2 Switches

The switches are indicated on page 6 as a ~ h (a ~ i for the electromagnetic brake type).

Information within the brackets [ ] refers to driver model UDK5107N only.

Switch Name	Indication	Factory setting	Function	Page Reference
a Motor Running Current Adjustment Rotary Switch	RUN	F	The motor running current can be adjusted with this digital switch. Adjustment is simple and an ammeter is not necessary.	Page 38, 39
b Motor Standstill Current Adjustment Rotary Switch	STOP	7	The motor standstill current can be reduced with this digital switch. Adjustment is simple and an ammeter is not necessary. Be sure to keep the switch set to 7 or below.	Page 38, 39
c Automatic Current Off Function Switch	AHO/OFF [O/OFF]	AHO [O]	This function will automatically cut the power to the motor when the internal temperature of the driver rises above 80°C. This function can be enabled or disabled with this switch.	Page 18, 19
d Step Angle Switch	FULL/HALF [F/H]	FULL [F]	The motor step angle can be set to full step or half step with this switch.	Page 18, 19
e Pulse Input Mode Switch	2P/1P [2/1]	2P [2]	The pulse signal input mode can be set to 1 pulse input mode or 2 pulse input mode with this switch.	Page 18, 20
f Self Test Function Switch	NORM/TEST [N/T]	NORM [N]	This function allows for verification of correct wiring connections between the motor and driver. The test can be enabled and disabled with this switch.	Page 18, 20
g Overheat Output Logic Switch	SNO/SNC [O/C]	SNO [O]	This switch sets the output logic for the overheat signal. SNO [O] : Normal open SNC [C] : Normal closed Match the setting to your equipment.	Page 18, 21
h Electromagnetic Brake Function Switch (For electromagnetic brake type only)	MBF/OFF	MBF	This switch sets the electromagnetic brake operation mode. MBF: Normally released, engaged when power is off OFF: Normally engaged, released through the M.B.FREE signal	Page 18, 21

### 3.3 Terminals

The input and output terminals are indicated on page 6 as ①~⑧ (①~⑩ for the electromagnetic brake type).

Information within the brackets [ ] refers to driver model UDK5107N only.

Terminal Name	Indication	Function	Page Reference
① CW Pulse Signal Input Terminal	CW	The CW direction command signal is input to this terminal. When a pulse is input to the terminal the motor output shaft will rotate one step in the clockwise direction. (When in 1 pulse input mode a pulse signal is input to this terminal.)	Page 22, 23
② CCW Pulse Signal Input Terminal	CCW	The CCW direction command signal is input to this terminal. When a pulse is input to the terminal the motor output shaft will rotate one step in the counterclockwise direction. (When in 1 pulse input mode a rotation direction signal is input to this terminal.)	Page 22, 23
③ Output Current Off Signal Input Terminal	H.OFF	The output current off signal is input to this terminal. When a signal is input to the terminal the driver will cut the power supply to the motor. The motor torque will then be reduced to zero and the motor shaft can be rotated freely for adjustment. This function is used when manually setting the motor to the home position etc.	Page 24
④ Excitation Timing Signal Output Terminal	TIMING [TIM]	The excitation timing signal is output from this terminal. This signal is output when the motor excitation (current running through the winding) is in the initial stage.	—
⑤ Overheat Signal Output Terminal	O.HEAT [O.H.]	The overheat signal is output from this terminal. This signal is output when the internal temperature of the driver rises above 80°C. This is used to prevent excess heat from damaging the driver.	Page 27, 28
⑥ Motor Connection Terminal	MOTOR [1, 2, 3, 4, 5]	This is the output terminal for the motor. The colors indicated on this terminal are matched to the motor lead wires for connection.	Page 30 ~ 32
⑦ Frame Ground Terminal	FG	This terminal is used to ground the driver case. Make a one point ground between this terminal and the controller FG terminal.	Page 34
⑧ Power Source Connection Terminal	AC100/115V	Connect this terminal to a power source of either single phase 100V ± 15% 50/60Hz or 115V ± 15% 60Hz.	Page 35
⑨ Electromagnetic Brake Release Signal Input Terminal (For electromagnetic brake type only)	M.B.FREE	The electromagnetic brake release signal is input to this terminal. Inputting this signal will release the electromagnetic brake. This terminal is used to release and engage the brake by means of an external signal.	Page 25, 26
⑩ Electromagnetic Brake Connection Terminal (For electromagnetic brake type only)	M.BRAKE	This is the output terminal for the electromagnetic brake. Connect it to the electromagnetic brake.	Page 30 ~ 32

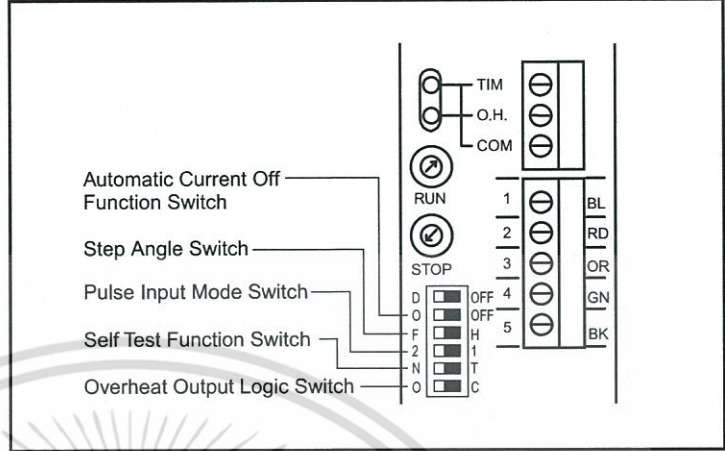
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# 5. Driver Function Switches

The driver has various operation functions which are set with the function switches.

## Driver Front Panel

**Standard Type Driver: UDK5107N**



**Standard Type Driver: UDK5114N**  
**High Speed Type Driver: UDK5128N**

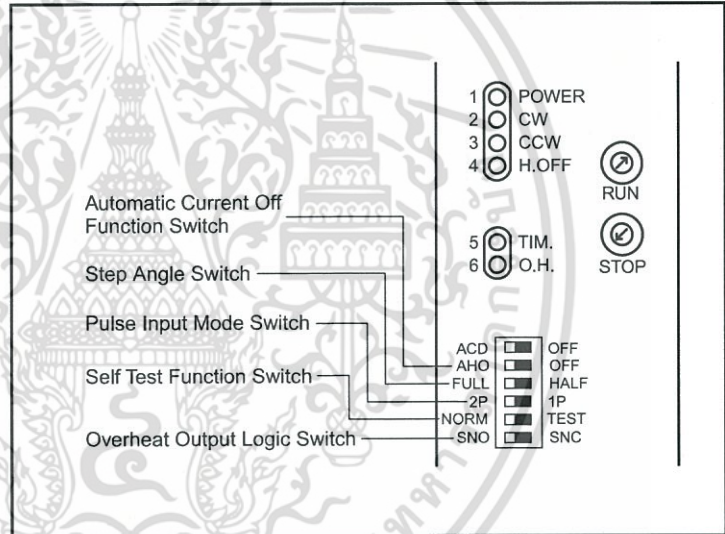


Illustration shows UDK5114N.

**Electromagnetic Brake**  
**Standard Type Driver: UDK5114N-M**  
**Electromagnetic Brake**  
**High Speed Type Driver: UDK5128N-M**

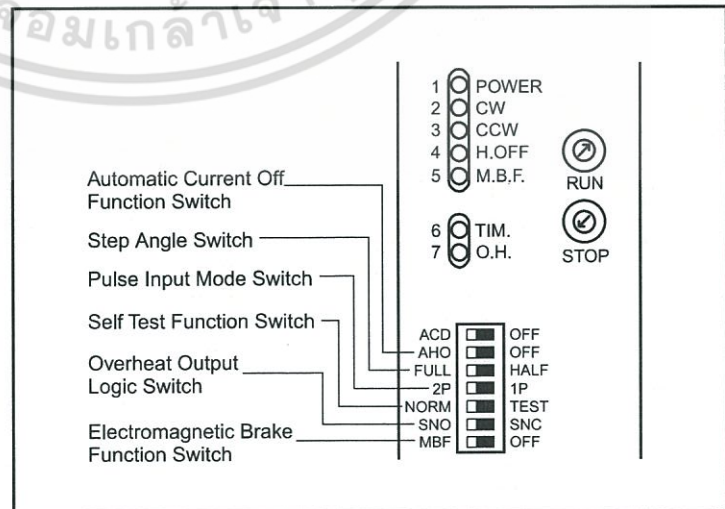
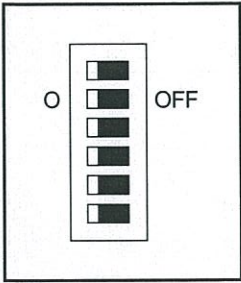


Illustration shows UDK5114N-M.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ The white square section of the function switch represents the switch lever.

## 5.1 Automatic Current Off Function Switch

UDK5107N

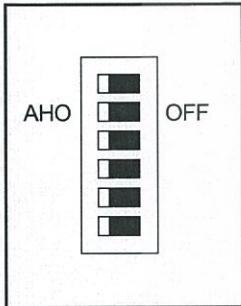


(Factory Setting: AHO [O])

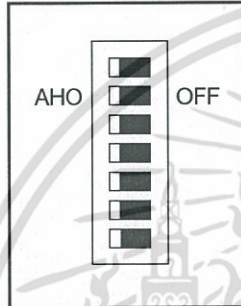
information within the brackets [ ] refers to driver model UDK5107N only.

When the automatic function switch is set to the AHO [O] position, the automatic current off function is enabled. While enabled, if the internal temperature of the driver rises above 80°C, the overheat signal will be output, and the current to the motor will be cut off. (Refer to pages 27, 28 for details on the overheat signal.) Cutting off the current to the motor will prevent driver heat damage. When the switch is set to the OFF position, the automatic current off function is disabled.

UDK5114N  
UDK5128N

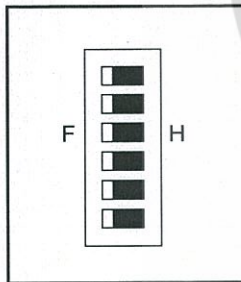


UDK5114N-M  
UDK5128N-M



## 5.2 Step Angle Switch

UDK5107N



(Factory Setting: FULL [F])

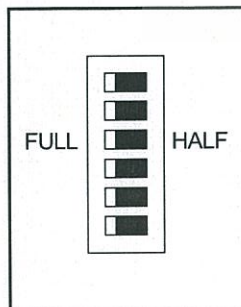
information within the brackets [ ] refers to driver model UDK5107N only.

When the switch is set to:

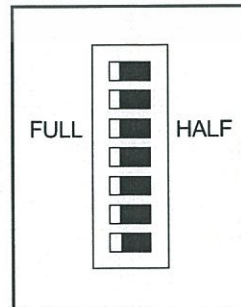
FULL [F] → 1 step = 0.72° (1 rotation = 500 pulses)

HALF [H] → 1 step = 0.36° (1 rotation = 1000 pulses)

UDK5114N  
UDK5128N

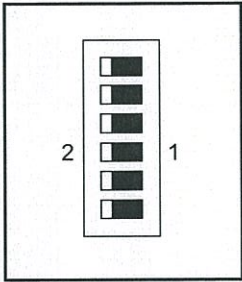


UDK5114N-M  
UDK5128N-M



### 5.3 Pulse Input Mode Switch

UDK5107N



(Factory Setting: 2P [2])

information within the brackets [ ] refers to driver model UDK5107N only.

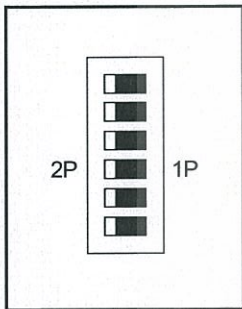
Select the appropriate pulse input mode to correspond to your controller with this switch.

When the pulse input mode switch is set to the 2P [2] position, 2 pulse input mode is established and motor rotation is controlled by CW and CCW pulse signals.

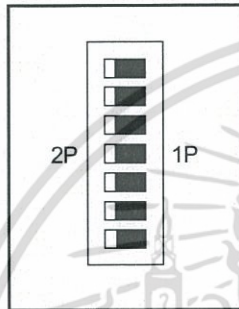
When the switch is set to the 1P [1] position, 1 pulse input mode is established and motor rotation is controlled by pulse signals and rotation direction (CW/CCW) signals.

(Refer to pages 22, 23 for a detailed explanation.)

UDK5114N  
UDK5128N

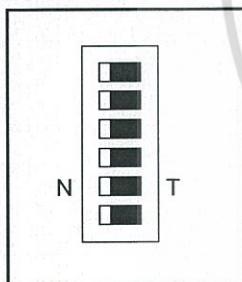


UDK5114N-M  
UDK5128N-M



### 5.4 Self Test Function Switch

UDK5107N



(Factory Setting: NORM [N])

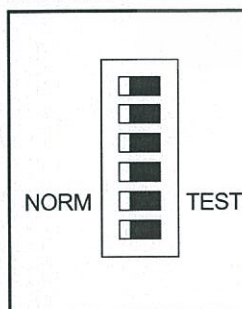
information within the brackets [ ] refers to driver model UDK5107N only.

When the self test function switch is set to the TEST [T] position, the self test function is activated. The self test is used to verify that the connections between the motor and driver are correct. (For instructions, refer to page 36, "Executing the Self Test Function".)

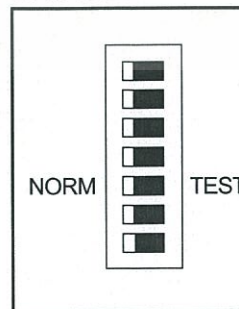
When the self test function switch is set to the NORM [N] position, the self test function is disabled.

During normal operation be sure to keep the switch set to the NORM [N] position.

UDK5114N  
UDK5128N



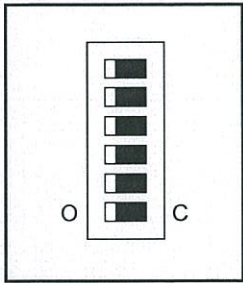
UDK5114N-M  
UDK5128N-M



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.5 Overheat Output Logic Switch

UDK5107N



(Factory Setting: SNO [O])

information within the brackets [ ] refers to driver model UDK5107N only.

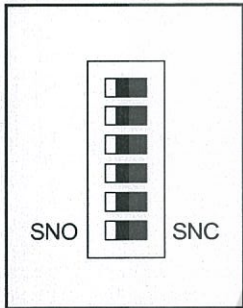
When the overheat output logic switch is set to the SNO [O] position, “H” level (photocoupler OFF) is the normal condition, and “L” level (photocoupler ON) is the condition when the overheat signal is output.

When the overheat output logic switch is set to the SNC [C] position, “L” level (photocoupler ON) is the normal condition, and “H” level (photocoupler OFF) is the condition when the overheat signal is output.

(For details on the overheat signal refer to pages 27, 28)

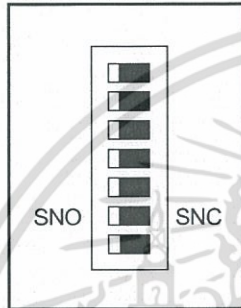
UDK5114N

UDK5128N



UDK5114N-M

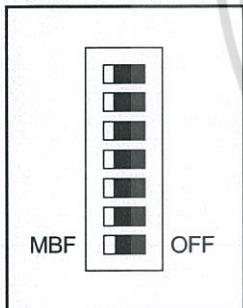
UDK5128N-M



## 5.6 Electromagnetic Brake Function Switch

UDK5114N-M

UDK5128N-M



(Factory Setting: MBF)

When the electromagnetic function switch is set to the MBF position, the electromagnetic brake is released (free) under normal conditions. If the driver power is cut off by a power failure etc., the brake will engage and hold the motor and load in position.

When the switch is set to the OFF position the electromagnetic brake is engaged, and the motor shaft is held in position. To release the brake for motor operation, input the electromagnetic brake release signal. (For instructions refer to pages 25, 26, “Electromagnetic Brake Release Signal”)

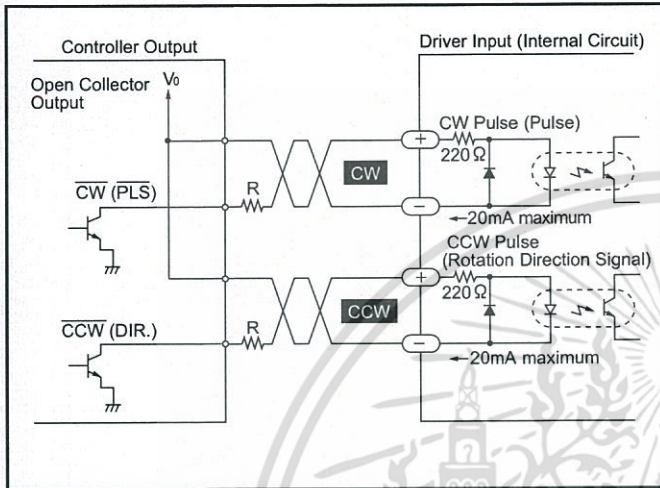
## 6. Input / Output Signals

### 6.1 Input Signals

The input signals to the driver and their functions are specified below.

#### 6.1.1 CW Pulse (CW) / Pulse (PLS) Signals CCW Pulse (CCW) / Rotation Direction (DIR.) Signals

The diagram below shows the input circuits and an example connection to a controller.



■ indicates the terminals as they appear on the front panel of the driver.

The information in the brackets ( ) refers to signals when in 1 pulse input mode.

Keep the voltage between DC5V and DC24V.

When voltage is equal to DC5V, external resistance  $R$  is not necessary.

When voltage is above DC5V, connect external resistance  $R$  and keep the input current below 20mA.

#### 2 Pulse Input Mode

##### CW\* pulse signal

When a negative logic pulse is input to the CW pulse signal input terminal, the motor rotates one step in the clockwise direction on the pulse rising edge.

##### CCW\* pulse signal

When a negative logic pulse is input to the CCW pulse signal input terminal, the motor rotates one step in the counterclockwise direction on the pulse rising edge.

\* CW and CCW refer to clockwise and counterclockwise directions respectively, from a reference point of facing the motor output shaft.

#### 1 Pulse Input Mode

##### Pulse signal

When a negative logic pulse is input to the CW pulse signal input terminal, the motor rotates one step on the pulse rising edge.

The direction of rotation is determined by the following rotation direction signals.

##### Rotation direction signal

The rotation direction signal is input to CCW pulse input terminal.

An "L" level signal input (photocoupler ON) commands clockwise direction rotation.

An "H" level signal input (photocoupler OFF) commands counterclockwise direction rotation.

#### Relation to the CW Pulse Signal LED (See pages 6, 7)

The LED lights when a CW pulse signal is input.

(In 1 pulse input mode, the LED indicates input of a pulse signal.)

#### Relation to the CCW Pulse Signal LED (See pages 6, 7)

The LED lights when a CCW pulse signal is input.

(In 1 pulse input mode, the LED indicates input of a rotation direction signal.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Relation to the Pulse Input Mode Switch (See pages 18, 20)

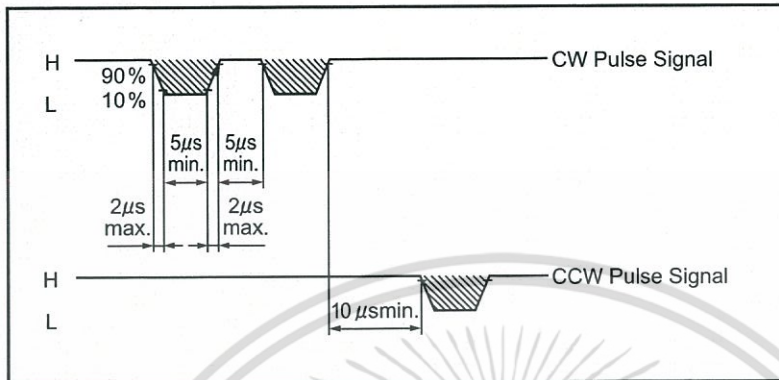
information within the brackets [ ] refers to driver model UDK5107N only.

When the switch is set to the 2P [2] position, motor rotation is controlled by CW pulse signals and CCW pulse signals.

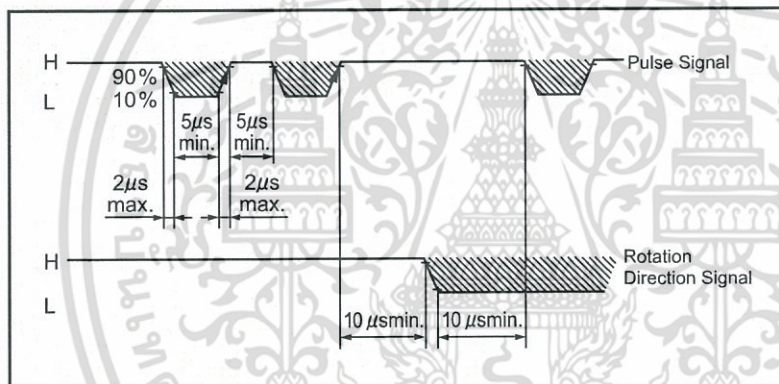
When the switch is set to the 1P [1] position, motor rotation is controlled by pulse signals and rotation direction signals.

### Pulse Waveform Characteristics

#### 2 Pulse Input Mode



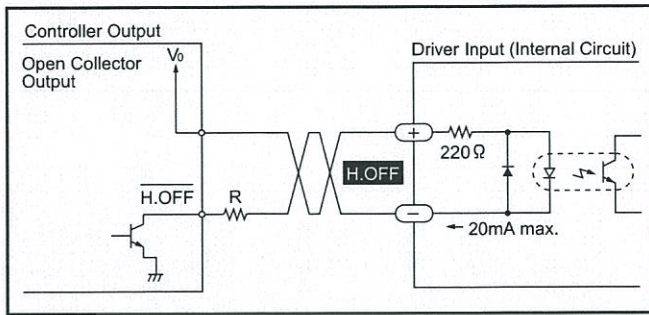
#### 1 Pulse Input Mode



- The shaded area indicates when the photocoupler diode is ON. The motor moves on the pulse rising edge as indicated by the arrow.
- The pulse voltage is “H” level = 4 ~ 5V, “L” level = 0 ~ 0.5V.
- Input pulse signals should have a pulse width over 5 μ, pulse rise/fall below 2 μ, and a pulse duty below 50%.
- Keep the pulse signal at “H” level when no pulse is being input, otherwise the automatic current cutback function will not be activated.
- The minimum interval time when changing rotation directions is 10 μ .  
This value varies greatly depending on the motor type, pulse frequency, and load inertia. It may be necessary to increase this time interval.
- When in 2 pulse input mode, do not input CW and CCW pulse signals at the same time. Inputting a pulse signal while the other pulse signal is already at “L” level will result in erratic motor rotation.
- When in 1 pulse input mode, leave the pulse signal at rest (“H” level) when changing rotation directions.

### 6.1.2 Output Current Off (H.OFF) Signal

The diagram below shows the input circuit and an example connection to a controller.



■ indicates the terminals as they appear on the front panel of the driver.

Keep the voltage between DC5V and DC24V.  
 When voltage is equal to DC5V, external resistance R is not necessary.  
 When voltage is above DC5V, connect external resistance R, and keep the input current below 20mA.

When the H.OFF signal is at “L” level (photocoupler ON), the current to the motor is cut off and the motor torque is reduced to zero. The motor output shaft can then be rotated freely by hand.

When the H.OFF signal is at the “H” level (photocoupler OFF), the motor holding torque is proportional to the current set by the current adjustment rotary switches. During motor operation be sure to keep the signal at “H” level.

This signal is used when moving the motor by external force, and for manual home positioning etc. If this function is not needed, it is not necessary to connect this terminal.

Switching the H.OFF signal from “L” level to “H” level does not alter the excitation sequence.

When the motor shaft is manually adjusted with the H.OFF signal input, the shaft will shift up to  $\pm 3.6^\circ$  from the position set after the H.OFF signal is released.

#### Relation to the Output Current Off Signal LED (See pages 6, 7)

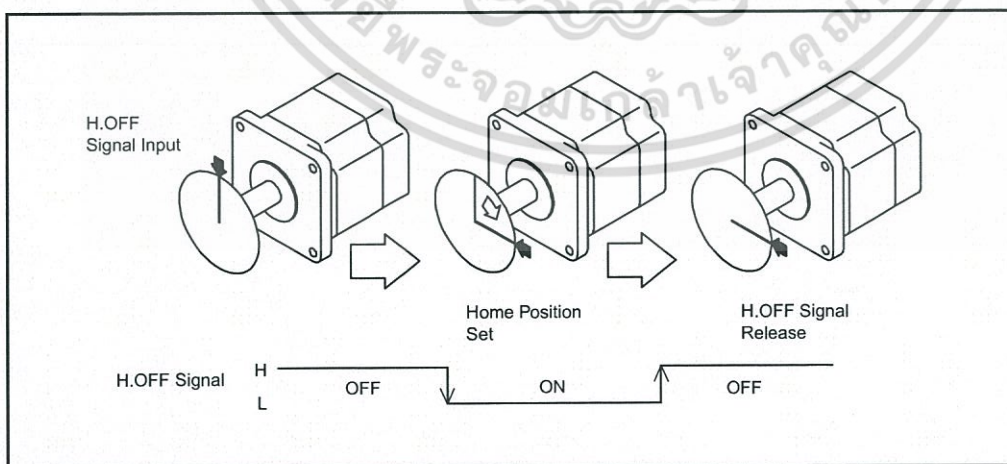
The LED lights when the H.OFF signal is input.

#### When Using the Electromagnetic Brake Type (See pages 18, 21)

Release the electromagnetic brake when inputting the H.OFF signal. To release the brake, either set the electromagnetic brake function switch to the MBF position, or set the switch to the OFF position and input (“L” level, photocoupler ON) the M.B. FREE signal.

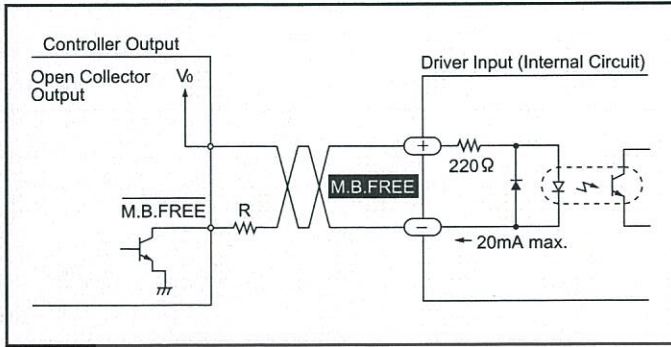
#### Manual Detection of the Home Position

Input the H.OFF signal, set the motor to the desired position, then release the H.OFF signal.



### 6.1.3 Electromagnetic Brake Release (M.B.FREE) Signal

The diagram below shows the input circuit and an example connection to a controller.



■ indicates the terminals as they appear on the front panel of the driver.

Keep the voltage between DC5V and DC24V.

When voltage is equal to DC5V, external resistance R is not necessary.

When voltage is above DC5V, connect external resistance R and keep the input current below 20mA.

When the M.B.FREE signal is at “L” level (photocoupler ON), the electromagnetic brake is released and the motor is ready for operation.

When the M.B.FREE signal is at the “H” level (photocoupler OFF), the electromagnetic brake is engaged and the motor shaft is held in position.

When the motor is at rest (pulse signals at rest), using the H.OFF signal while the motor is held in position with the electromagnetic brake allows for a reduction in motor heat generation and power consumption.

#### Relation to the Electromagnetic Brake Release Signal LED (See pages 6, 7)

The LED lights when the M.B.FREE signal is input.

#### Relation to the Electromagnetic Brake Function Switch (See pages 18, 21)

When the switch is set to the OFF position, the brake is engaged and released through the M.B.FREE signal.

When the switch is set to the MBF position, the brake is only engaged when the driver power is OFF, and cannot be released through a signal. (The M.B.FREE signal is not valid.)

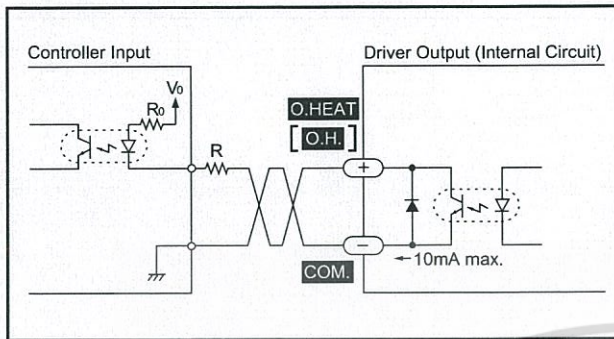
When using the M.B.FREE signal, be sure to keep the switch set the OFF position.

## 6.2 Output Signals

The output signals to the driver and their functions are specified below.

### 6.2.1 Overheat (O.HEAT) Signal

The diagram below shows the output circuit and an example connection to a controller.



■ indicates the terminals as they appear on the front panel of the driver.

The information in the brackets [ ] refers to driver model UDK5107N only.

Keep the voltage between DC5V and DC24V.

Keep the current below 10mA.

If the current exceeds 10mA, connect external resistance R.

The O.HEAT signal is output to protect the driver from heat damage when the internal temperature of the driver rises above 80°C.

When the O.HEAT signal is output, turn the driver power OFF, then adjust the operating conditions (ambient temperature, driver/controller settings, etc.), or use a fan etc. to cool the driver.

#### Relation to the Overheat Signal Output LED (See pages 6, 7)

The LED lights when the O.HEAT signal is output.

#### Relation to the Overheat Output Logic Switch (See pages 18, 21)

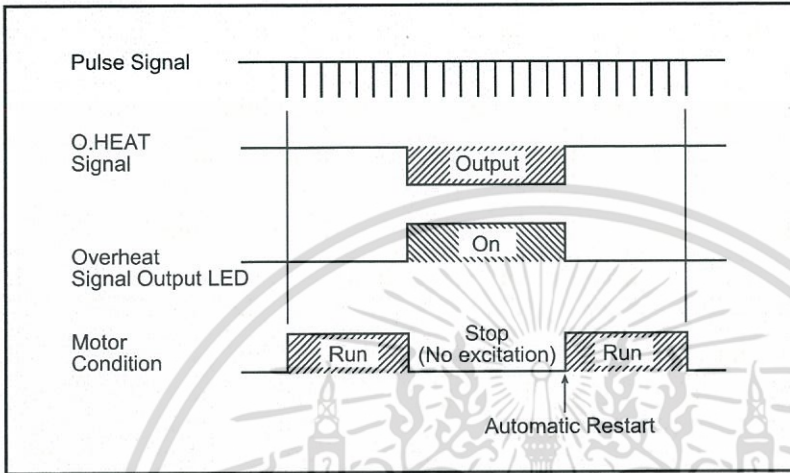
When the switch is set to the SNO [O] position, the O.HEAT signal is output as an “L” level signal (photocopler ON).

When the switch is set to the SNC [C] position, the O.HEAT signal is output as an “H” level signal (photocopler OFF).

**Relation to the Automatic Current Off Function Switch (See pages 18, 19)**

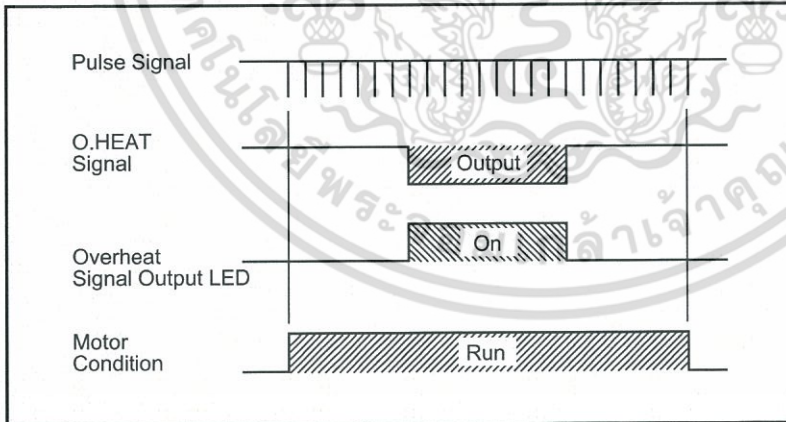
**When set to AHO [O]**

- ① The O.HEAT signal is output when the internal temperature of the driver exceeds 80°C during operation.
- ② Regardless of any pulse signals input, motor excitation ceases (shaft becomes free) and the motor comes to a natural stop.
- ③ After operation stops, when the driver internal temperature returns to below 80°C the O.HEAT signal is released.
- ④ Operation restarts (automatic restart)



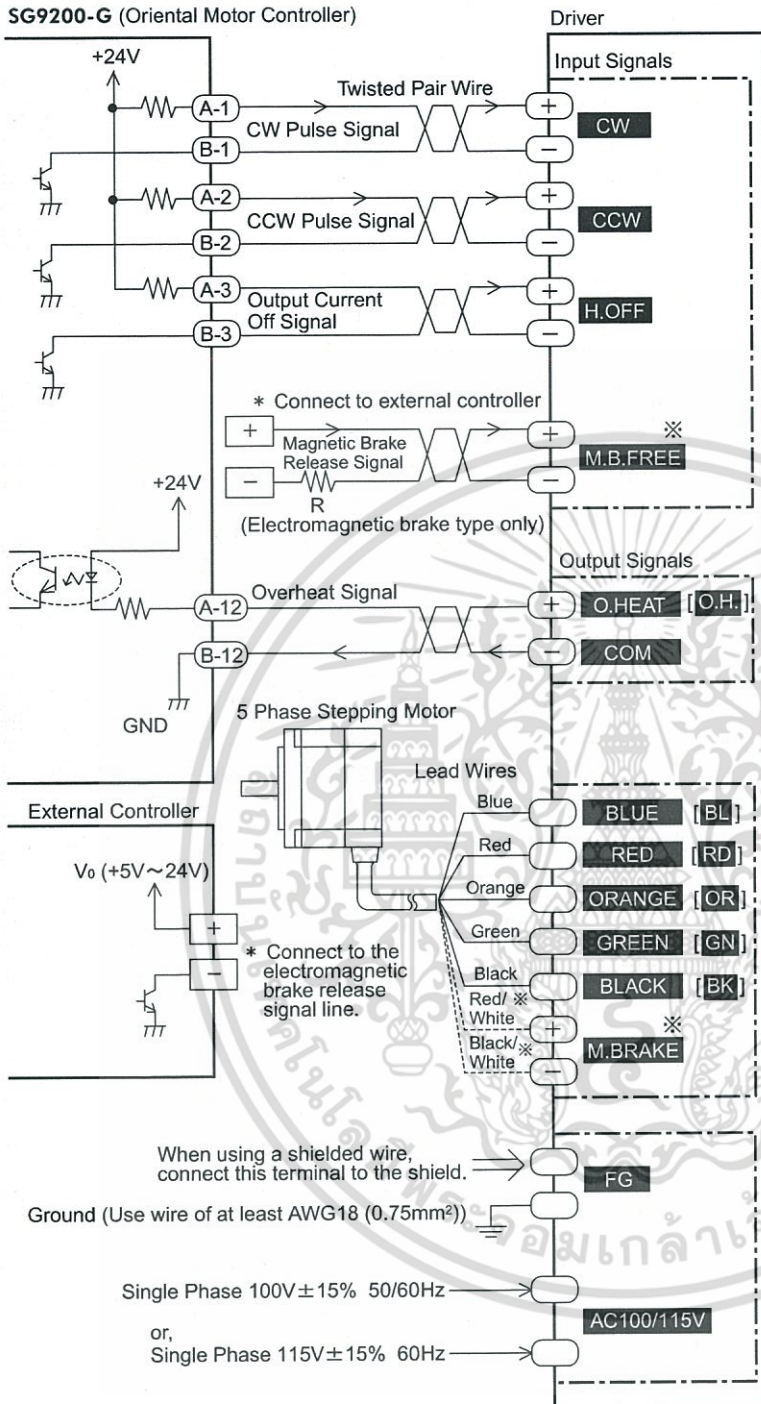
**When set to OFF**

- ① The O.HEAT signal is output when the internal temperature of the driver exceeds 80°C during operation.
- ② The motor will continue to run regardless of the O.HEAT signal output.
- ③ When the driver internal temperature returns to below 80°C the O.HEAT signal is released.
- ④ Operation continues



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**SG9200-G (Oriental Motor Controller)**



■ indicates the terminals as they appear on the front panel of the driver.

Information within the brackets [ ] refers to driver model UDK5107N only.

□ indicates the combination connector.

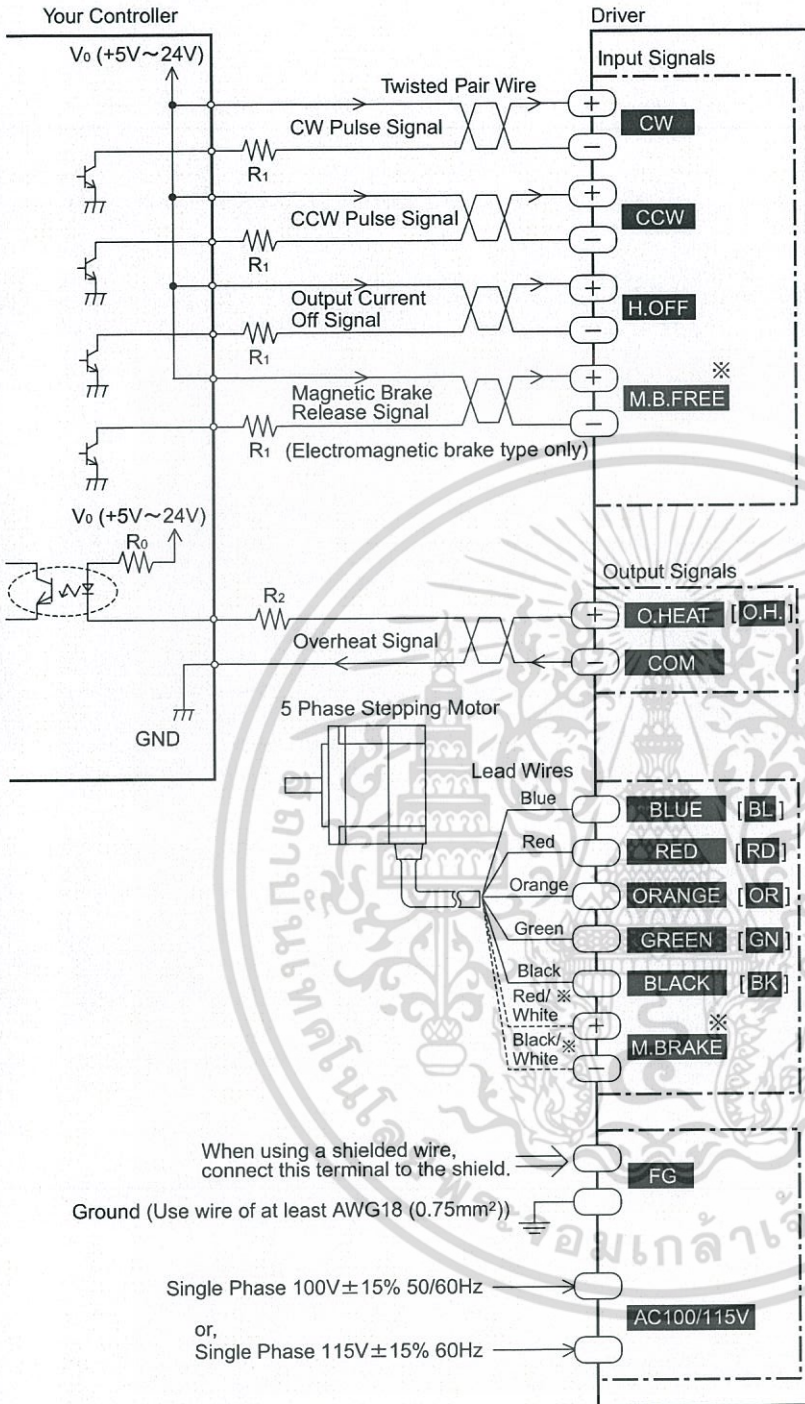
The numbers within ○ on the SG9200-G indicate the pin number.

※ indicates electromagnetic brake type only.

**※For the electromagnetic brake release signal connections**

Keep the voltage between DC5V and DC24V. When voltage is equal to DC5V, external resistance R is not necessary. When voltage is above DC5V, connect external resistance R and keep the input current below 20mA.

### 7.1.2 Connections to Your Controller



■ indicates the terminals as they appear on the front panel of the driver.

Information within the brackets [ ] refers to driver model UDK5107N only.

⎓ indicates the combination connector.

※ indicates electromagnetic brake type only.

**For Input signal connections:**

Keep the voltage between DC5V and DC24V. When voltage is equal to DC5V, external resistance R<sub>1</sub> is not necessary. When voltage is above DC5V, connect external resistance R<sub>1</sub> and keep the input current below 20mA.

**For output signal connections:**

Keep the voltage between DC5V and DC24V. Keep the current below 10mA. If the current exceeds 10mA, connect external resistance R<sub>2</sub>.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

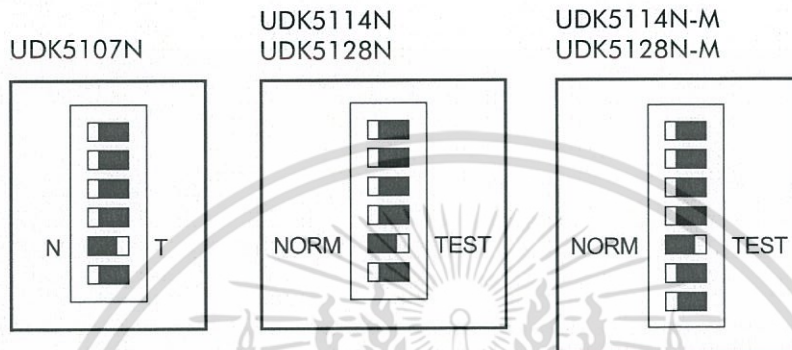
## 7.8 Executing the Self Test Function

The self test function has been incorporated into the driver to allow for verification of correct connections between the motor and driver.

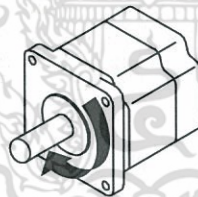
To execute the self test function follow the procedure below.

Information within brackets [ ] refers to driver model UDK5107N only.

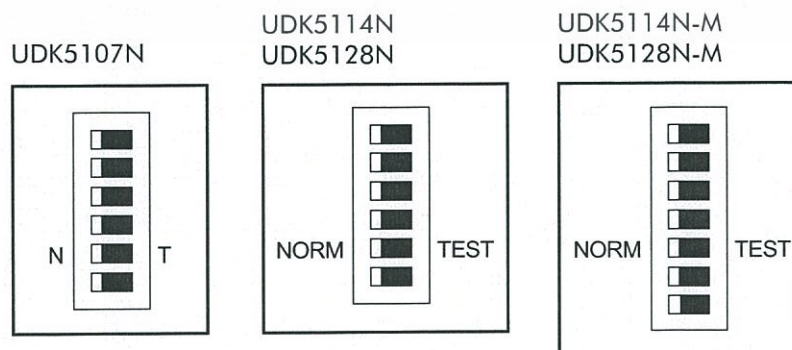
1. With the motor secured to the mounting plate, remove the coupling, etc. and disconnect the load.
2. Connect the motor to the driver.
3. After confirming that all the input/output signals are disconnected, turn the power ON.
4. Set the self test function switch to the TEST [T] position.



5. When the pulse mode switch is set to 2P [2], the motor should begin to rotate clockwise (counterclockwise when the pulse mode switch is set to 1P [1]) at a pulse speed of approximately 3Hz immediately after the self test switch is set to the TEST [T] position. This condition of operation indicates that all connections between the motor and driver are correct. However, if the motor shaft rotates in abrupt movements, or rotates in the opposite direction, connections are abnormal. Turn the power OFF immediately and check the connections. The motor will continue to rotate while the self test function switch is set to the TEST [T] position.



6. Return the self test switch from the TEST [T] position to the NORM [N] position. The self test is now complete. The motor will stop and the driver will be returned to normal operation mode. Be sure to keep the switch set to NORM [N] during operation.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

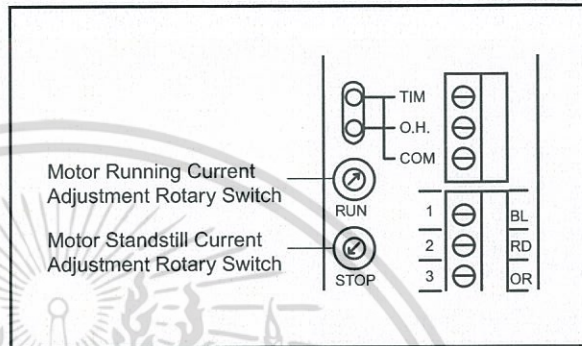
## 8. Motor Current Adjustment

If maximum motor torque is not needed, adjustment of the motor running current or the motor standstill current can be used to reduce motor vibration and motor and driver heat generation.

- To reduce temperature rise of the motor and driver → Reduce the motor running current and the motor standstill current
- To reduce motor vibration due to excess torque → Reduce the motor running current

### Driver Front Panel

Standard Type Driver: UDK5107N



Standard Type Driver: UDK5114N  
High Speed Type Driver: UDK5128N

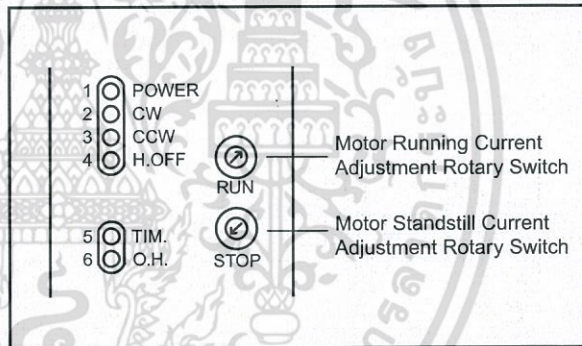


Illustration shows UDK5114N

Electromagnetic Brake  
Standard Type Driver: UDK5114N-M

Electromagnetic Brake  
High Speed Type Driver: UDK5128N-M

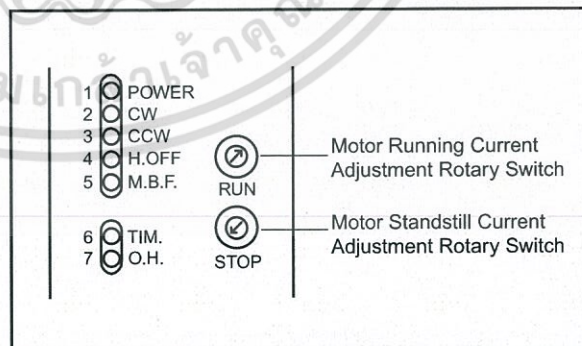


Illustration shows UDK5114N-M

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 8.1 Motor Running Current Adjustment

The motor running current is factory set to the motor's rated current.

(Motor running current adjustment rotary switch RUN: F)

Adjust the motor running current by turning the RUN rotary switch with a small slot screwdriver.

The RUN switch settings and corresponding current values are listed in the following chart.

**RUN switch settings and corresponding current values (representative values)**



RUN Switch Settings	Running Current [A/phase]		
	UDK5107N	UDK5114N UDK5114N-M	UDK5128N UDK5128N-M
0	0.21	0.45	0.76
1	0.25	0.52	0.90
2	0.28	0.58	1.04
3	0.32	0.64	1.17
4	0.36	0.71	1.31
5	0.39	0.77	1.44
6	0.43	0.83	1.58
7	0.46	0.89	1.72
8	0.50	0.96	1.85
9	0.54	1.02	1.99
A	0.57	1.09	2.13
B	0.61	1.15	2.26
C	0.64	1.21	2.39
D	0.68	1.27	2.53
E	0.72	1.33	2.67
F	0.75	1.40	2.80

## 8.2 Motor Standstill Current Adjustment

The current at motor standstill can be adjusted to reduce motor/driver heat generation. The motor standstill current is factory set to approximately 40% of the rated current (standstill current adjustment switch "STOP" set to "7").

The current is automatically reduced approximately 0.1sec. after pulse signals stop. **Be sure to keep the switch set to 7 or below.**

The amount of current reduction is proportional to the setting of the motor running current.

$$\text{Standstill Current [A/phase]} = \frac{\text{Running Current Setting [A/phase]} \times \text{Standstill Current Setting [\%]}}{100}$$

**STOP switch settings and corresponding rate of current reduction (representative values)**



STOP Switch Settings	% of Running Current		
	UDK5107N	UDK5114N UDK5114N-M	UDK5128N UDK5128N-M
0	10	19	9
1	10	19	9
2	10	19	9
3	15	21	14
4	23	28	21
5	29	34	28
6	37	40	36
7	43	47	43
8	51	54	50
9	57	61	57
A	65	67	64
B	72	74	71
C	79	80	79
D	86	87	86
E	93	94	93
F	100	100	100

## 9. Troubleshooting

Consult the following chart if the motor is not functioning properly. If the motor is still not functioning properly after confirming the check points below, contact your nearest sales office as listed at the back of this manual.

Information within brackets [ ] refers to driver model UDK5107N only.

PROBLEM	CHECK POINTS	MEASURES
No excitation in the motor. (The motor has no holding torque and the shaft can be turned freely by hand)	1. Is the driver POWER LED ON? (If ON, condition is normal)	If the POWER LED is not ON, check if the power source is properly connected. Verify that AC100V or AC115V is being input.
	2. Is the driver output current off signal input LED OFF? (If OFF, condition is normal)	When the H.OFF (output current off) signal is input the output current off signal input LED lights and the motor loses all excitation (no holding torque). Return the H.OFF signal to "H" level.
	3. Is the driver overheat LED OFF? (If OFF, condition is normal)	The overheat LED lights when the O.HEAT signal is output. If the automatic current off function switch is set to the AHO position when this signal is output, the motor will lose all excitation (no holding torque). Refer to items 26 ~ 29 and take the necessary steps to prevent the overheat signal from being output.
	4. Are the driver and motor correctly connected?	Check the driver connection terminals. If the motor cable has been extended check the extension connection.
	5. Are the current adjustment rotary switches (RUN and STOP) set too low?	These rotary switches control the output current to the motor (refer to pages 38, 39). If they are set too low return them to the factory set positions.
Note: If the motor still has no torque after checking the above conditions, the driver is probably defective. After reconfirming that the current voltage and connections are correct, contact your nearest sales office for service.		
The motor does not rotate.	First check the 5 items above.	
The motor does not rotate when a pulse signal is input.	6. Are the pulse signal lines correctly connected? Are the pulse signal waveform characteristics correct?	Check the connections and pulse signal waveform characteristics (refer to pages 22, 23). Use a controller which is able to output a standard pulse signal.
	7. While in 2 pulse input mode (pulse input mode switch set to the 2P [2] position), are either the CW or CCW pulse signal inputs already at "L" level?	The motor will not rotate if a pulse signal is input when the other pulse signal is at "L" level. Be sure to keep the pulse signal at "H" level.
	8. While in 1 pulse input mode (pulse input mode switch set to the 1P [1] position), is the pulse signal connected to the CCW pulse signal terminal?	Connect the pulse signal to the CW pulse signal terminal.
	9. For the electromagnetic brake type, is the M.B.FREE signal (brake release) at "H" level while the electromagnetic brake function switch is set to OFF?	If the M.B.FREE signal is at "H" level, the brake is not released and the motor will not operate. Be sure to keep the M.B.FREE signal at "L" level during motor operation.
The motor rotates in the wrong direction.	10. While in 2 pulse input mode (pulse input mode switch set to the 2P [2] position) are the CW and CCW pulse signal lines connected backwards?	Connect the CW pulse signal line to the CW pulse signal input terminal and connect the CCW pulse signal line to the CCW pulse input terminal.
	11. While in 1 pulse input mode (pulse input mode switch set to the 1P [1] position) leave the CCW pulse signal terminal unconnected and try inputting a pulse signal to the CW pulse signal terminal.	If the motor rotates in a counterclockwise direction when a pulse signal is input, the motor and driver are normal. Recheck the rotation direction signal levels. ( "L" level = clockwise, "H" level = counterclockwise)

PROBLEM	CHECK POINTS	MEASURES
Motor rotation is erratic.	First check items 4, 5, and 6.	
Motor start up is unstable.	12. While In 2 pulse input mode (pulse input mode switch set to the 2P [2] position) are the driver's CW LED and CCW LED ON at the same time?	If both pulse signals are input at the same time motor operation will be unstable. Be sure to input only one pulse at a time.
	13. Are the motor shaft and load properly aligned? Is the load too heavy for the motor?	Make sure the motor shaft and load are securely attached and properly aligned. Recheck the operating conditions, and if necessary lighten the load.
The motor rotates too far or not far enough.	14. Does the step angle required by your equipment match the step angle of the stepping motor?	Check the setting of the step angle switch located on the driver.
	15. Is the number of pulses set to match the amount of motor rotation?	Check the controller pulse setting.
The motor loses synchronization during acceleration or while running.	16. Is the driver overheat signal output LED OFF? (If OFF, condition is normal)	The overheat signal output LED lights when the overheat signal is output. If the automatic output current off function switch is set to the AHO [O] position when this signal is output the motor will lose all excitation (no holding torque). Refer to items 26 ~ 29 and take the necessary steps to prevent the overheat signal from being output.
	17. Is the starting pulse frequency too high?	Check this by decreasing the frequency.
	18. Is the acceleration or deceleration time too short?	Check this by increasing the acceleration/deceleration time.
	19. Is the motor being affected by noise interference?	Check this by running the motor while the machine suspected of producing the noise interference is off.
Motor vibration is very high.	20. Is the output torque too high?	Try reducing the motor running current with the "RUN" current adjustment rotary switch.
	21. Try changing the pulse frequency.	If the vibration decreases after the pulse frequency has been adjusted, this means the motor is resonating. Either adjust the frequency or change the step angle. Also try installing the optional (sold separately) clean damper (for double shaft model only).
Motor temperature is very high.	22. Is the motor running time too long? (Is the temperature of the motor case below 100°C?)	Shorten the running time or increase the resting time. (The temperature of the motor may rise considerably depending on the operating conditions. During high speeds and depending on the duty drive cycle, the motor could be susceptible to heat damage. Allow for sufficient heat dissipation from the motor, and keep the temperature of the motor case below 100°C)
	23. Is the driver standstill current adjustment switch set to "8" or above?	Refer to pages 38, 39 and set the switch to "7" or below.
	24. Is the driver CW LED or CCW LED still ON after pulse signals are complete?	While the pulse signal is kept at "L" level the CW or CCW pulse LED remains ON, and the motor current is not reduced. Return the pulse signal to "H" level.

Information within brackets [ ] refers to driver model UDK5107N only.

PROBLEM	CHECK POINTS	MEASURES
The electromagnetic brake does not hold.	25. Is the electromagnetic brake function switch set to the MBF position while the M.B.FREE (brake release) signal is at "H" level?	Set the electromagnetic brake function switch to the OFF position and keep the M.B.FREE (brake release) signal at "H" level.
The overheat signal is output.	26. Is the driver ambient temperature 0°C~ 50°C?	If not, take the necessary steps to keep the ambient temperature within 0°C~ 50°C.
	27. Is the driver located in an enclosed or poorly ventilated area?	Install the driver in a well ventilated area, or install a ventilation fan.
	23. Is the driver mounted to a metal surface?	If not, mount the driver to a metal surface or install a ventilation fan.
	24. Is the driver continuously operating at a pulse rate which requires the maximum input current?	If changing the pulse rate is a possibility, try adjusting it enough to decrease the input current. For details refer to the driver input current indicated in the general catalog "speed vs. torque characteristics".



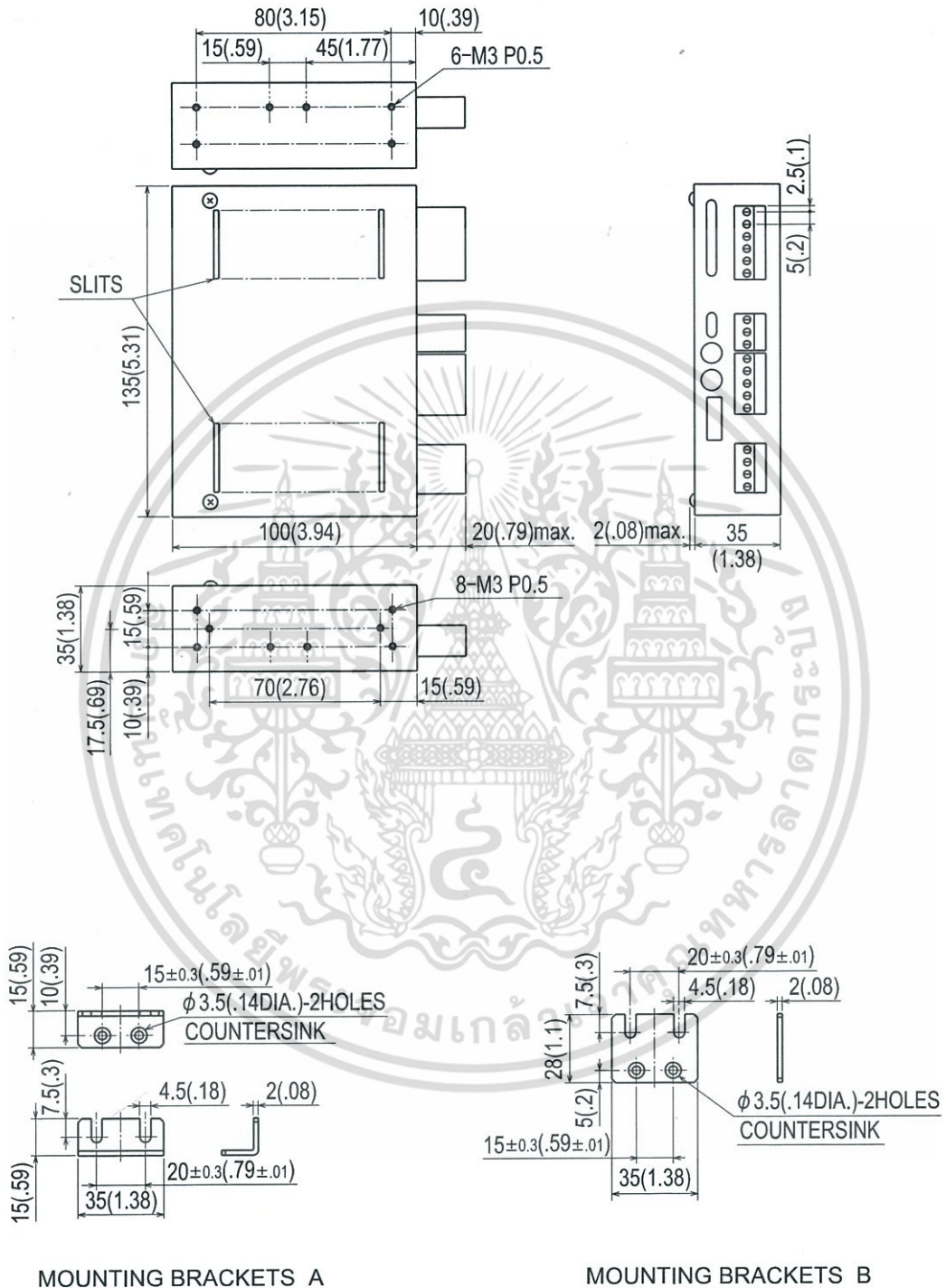
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านกรรค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 11.2 Driver

Unit: mm (inch)

### 11.2.1 Standard Type

Model: UDK5107N



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

