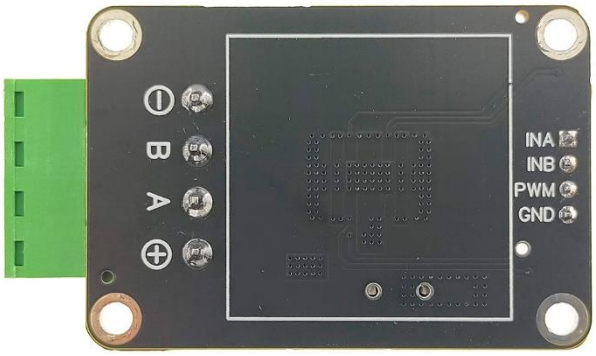
Pakkret

บอร์ดขับมอเตอร์ 5A/4-28V

- 5A (ต่อเนื่อง), 15A (ชั่วขณะ)
- 4-28V DC
- INA, INB, PWM
- CC, CCW, Break, Hi-z
- ระบบป้องกันครบ



Pakkret (ปากเกร็ด) เป็นบอร์ดขับมอเตอร์แปลงถ่าน (Brushed motor) / มอเตอร์ DC มีความสามารถขับมอเตอร์ด้วยกระแสสูงสุด 5A ต่อเนื่อง และ 15A ชั่วขณะ รองรับการควบคุมทิศทางและความเร็วของมอเตอร์ผ่านขา INA, INB, PWM หรือ INA, INB รับแรงดันไฟฟ้าเข้าได้กว้าง 4 - 28V ใช้ไฟเลี้ยงจากแบตเตอรี่ลิเธียมได้ 2 ถึง 6 เซลล์ มาพร้อมวงจรป้องกันจ่ายไฟพิดขั้ว ป้องกันกระแสเกิน ป้องกันความร้อนสูงเกิน ใช้งานร่วมกับบอร์ด Arduino ได้ง่าย ใช้ต่อขยายกำลังขับของบอร์ดควบคุมหุ่นยนต์ได้ เช่น ต่อขยายกำลังขับมอเตอร์ของบอร์ดของ INEX ทุกรุ่น, บอร์ดของ PrinceBot ทุกรุ่น, บอร์ดของ Designed By Sopon ทุกรุ่น, บอร์ดของ อาร์ทรอน ทุกรุ่น เป็นต้น

เวอร์ชั่น	วันที่เผยแพร่	รายละเอียดการแก้ไข
V1.0	13/6/2567	เผยแพร่ครั้งแรก
V1.1	25/6/2567	เพิ่มรูปการต่อใช้งานร่วมกับ POP-32i / ATX2+ แบบครบทุกมอเตอร์

สารบัญ

ส่วนประกอบและขาต่อใช้งาน 3

คุณสมบัติทางไฟฟ้า 4

โหมดการทำงาน 4

1) ควบคุมทิศทางด้วย INA, INB และควบคุมความเร็วด้วย PWM 4

2) ควบคุมทิศทางและความเร็วด้วย INA, INB 5

ระบบป้องกัน 5

ระบบป้องกันจ่ายไฟผิดขั้ว 5

ระบบป้องกันอุณหภูมิสูงเกิน 5

ระบบป้องกันกระแสไฟฟ้าเกิน 5

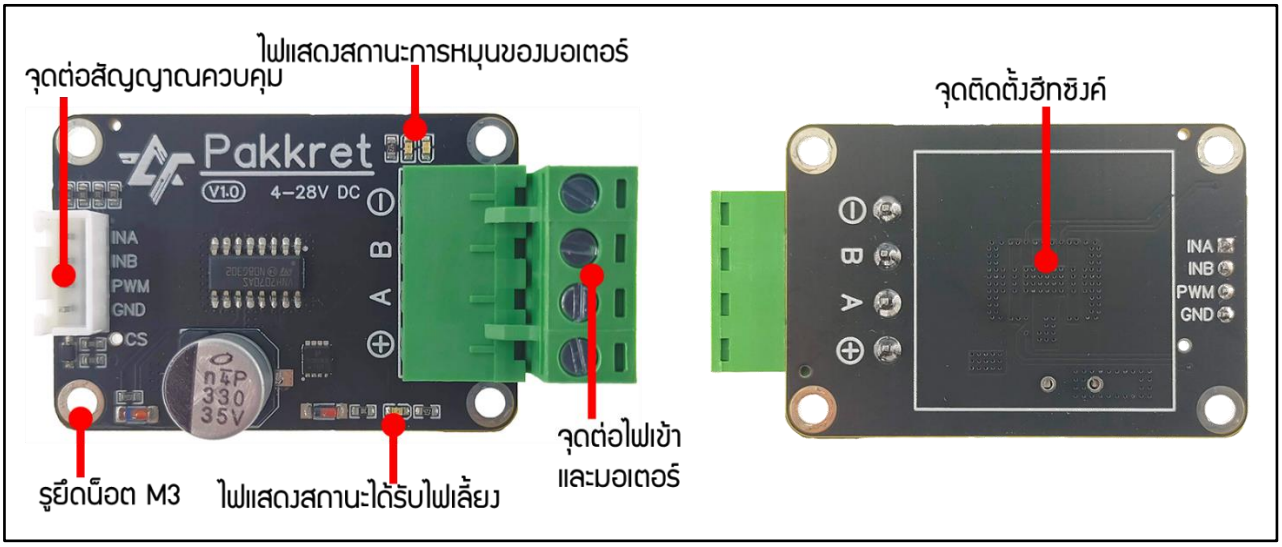
ระบบป้องกันแรงดันต่ำเกิน 6

การใช้งานร่วมกับบอร์ด Arduino 6

การใช้ขยายกำลังขับมอเตอร์ของบอร์ดควบคุมหุ่นยนต์ 9

ข้อแนะนำการใช้งาน 14

ส่วนประกอบและขาต่อใช้งาน



รูปที่ 1 ส่วนประกอบของบอร์ดขับมอเตอร์ ปากเกร็ด

ตารางที่ 1 ขาต่อใช้งาน

ชื่อขา	อินพุต / เอาต์พุต	รายละเอียด
INA	อินพุต	จุดต่อสัญญาณควบคุมทิศทางของมอเตอร์ 1 (กรณีใช้ขา PWM) และควบคุมความเร็วมอเตอร์ (กรณีไม่ใช้ขา PWM)
INB	อินพุต	จุดต่อสัญญาณควบคุมทิศทางของมอเตอร์ 2 (กรณีใช้ขา PWM) และควบคุมความเร็วมอเตอร์ (กรณีไม่ใช้ขา PWM)
PWM	อินพุต	จุดต่อสัญญาณ PWM (ปล่อยลอยได้ จะมีสถานะเป็นลอจิก 1 ค้าง)
GND		
+	อินพุต	จุดต่อไฟเลี้ยงมอเตอร์ขั้วบวก / จุดต่อแบตเตอรี่ขั้วบวก
A	เอาต์พุต	จุดต่อมอเตอร์ 1
B	เอาต์พุต	จุดต่อมอเตอร์ 2
-	อินพุต	จุดต่อไฟเลี้ยงมอเตอร์ลบ / จุดต่อแบตเตอรี่ขั้วลบ

คุณสมบัติทางไฟฟ้า

ตารางที่ 2 คุณสมบัติทางไฟฟ้า

รายการ	ต่ำสุด	แนะนำ	สูงสุด
แรงดันไฟฟ้าขาเข้า (V_{IN})	4V	12V / 24V	28V
แรงดันไฟฟ้าขาออก (V_{OUT})		V_{IN} x 0.98	
กระแสไฟฟ้าที่ใช้ (เมื่อ PWM = 0)			
กำลังขับต่อเนื่อง		5A	
กำลังขับชั่วขณะ			15A
แรงดันลอจิก 1 (HIGH)	2.1V		
แรงดันลอจิก 0 (LOW)			0.9V

อ่านคุณสมบัติทางไฟฟ้าฉบับเต็มได้ที่ VNH7070AS Datasheet

โหมดการทำงาน

การป้อนลอจิกเพื่อควบคุมมอเตอร์ทำได้ 2 แบบ ดังนี้

1) ควบคุมทิศทางด้วย INA, INB และควบคุมความเร็วด้วย PWM

การควบคุมรูปแบบนี้ใช้ขาดิจิทัลของไมโครคอนโทรลเลอร์ 2 ขา และใช้ขา PWM เพียง 1 ขาเท่านั้น โดยการป้อนลอจิก 0/1 เข้าที่ขา INA, INB จะเป็นการกำหนดทิศทางการหมุนของมอเตอร์ และการป้อนสัญญาณ PWM เข้าที่ขา PWM จะเป็นการกำหนดความเร็วการหมุนของมอเตอร์ โดยยิ่งกำหนดค่าดิวตี้ไซเคิล (Duty Cycle) มากที่ขา PWM จะทำให้มอเตอร์หมุนเร็วมาก หากกำหนดค่าดิวตี้ไซเคิลที่น้อย จะทำให้มอเตอร์หมุนช้าลง-หยุดหมุน

ตารางที่ 3 โหมดการทำงานเมื่อใช้การควบคุมแบบ INA, INB, PWM

INA	INB	PWM	เอาต์พุต
1	1	1	หยุดอย่างรวดเร็ว
1	0	1 / PWM	หมุนตรง
0	1	1 / PWM	หมุนกลับ
x	x	0	หยุดตามแรงเฉื่อย

2) ควบคุมทิศทางและความเร็วด้วย INA, INB

การควบคุมรูปแบบนี้ใช้ขา PWM ของไมโครคอนโทรลเลอร์ 2 ขา ควบคุมทิศทางและความเร็วของมอเตอร์ โดยยิ่งขาที่คุม PWM มีค่าดิวตี้ไซเคิลที่มาก จะยิ่งทำให้มอเตอร์หมุนช้าลง-หยุดหมุน หากกำหนดค่าดิวตี้ไซเคิลที่น้อย จะทำให้มอเตอร์หมุนเร็วขึ้น

ตารางที่ 4 โหมดการทำงานเมื่อใช้การควบคุมแบบ INA, INB

INA	INB	เอาต์พุต
1	1	หยุดอย่างรวดเร็ว
1	PWM	หมุนตรง
PWM	1	หมุนกลับ
0	0	หยุดตามแรงเฉื่อย

ระบบป้องกัน

ระบบป้องกันจ่ายไฟผิดขั้ว

ระบบออกแบบมาให้ใช้มอตเฟตชนิด P ในการป้องกันจ่ายไฟผิดขั้ว โดยใช้คุณสมบัติของมอตเฟต P นำกระแสเมื่อได้รับไฟเลี้ยงลบเข้าที่ขา Gate (G) หากมีการต่อถูกขั้ว ไฟเลี้ยงที่เข้าขา G จะเป็นลบ ทำให้มอตเฟตนำกระแส จ่ายไฟไปยังวงจรควบคุมมอเตอร์ แต่หากมีการต่อสลับขั้ว ไฟเลี้ยงที่เข้าขา G เป็นบวก มอตเฟตไม่นำกระแส ทำให้ไม่มีกระแสไฟฟ้าไหลผ่านไปยังวงจรควบคุมมอเตอร์ ช่วยป้องกันความเสียหายในวงจรควบคุมมอเตอร์ได้

ข้อจำกัดของวงจรส่วนนี้คือทนกระแสไฟฟ้าได้ไม่เกิน 30A/29V เมื่อมีการนำกระแสที่มาก จะยิ่งทำให้มอตเฟต P มีอุณหภูมิที่สูงขึ้นเนื่องจากการสูญเสียภายใน โดย อาร์ทรอน ได้ทดสอบแล้วว่า ที่กระแสไฟฟ้า 5A ต่อเนื่อง มอตเฟตไม่สร้างความร้อนจนรู้สึกได้ ดังนั้นมอตเฟต P จึงไม่จำเป็นต้องติดฮีทซิ้งค์เพิ่มเติมเมื่อใช้งานจริง

ระบบป้องกันอุณหภูมิสูงเกิน

เมื่อมีการนำกระแสไฟฟ้าสูงมาก ความร้อนภายในที่เกิดจากการสูญเสียจะยิ่งมีมากขึ้น โดยระบบป้องกันอุณหภูมิเกินจะป้องกันไม่ให้ความร้อนการสูญเสียไปทำลายโครงสร้างของซิลิคอนภายในชิปขับมอเตอร์ โดยระบบจะตัดการทำงานเมื่ออุณหภูมิ ณ แผ่นซิลิคอนอยู่ในช่วง 150 ถึง 200 องศาเซลเซียส และจะกลับมาทำงานได้อีกครั้งเมื่ออุณหภูมิลดลงต่ำกว่า 135 องซาเซลเซียส

ระบบป้องกันกระแสไฟฟ้าเกิน

กระแสไฟฟ้าที่สูงเกินจะทำลายลายวงจรภายในชิปขับมอเตอร์ โดยวงจรป้องกันกระแสไฟฟ้าเกินจะช่วยป้องกันไม่ให้กระแสไฟฟ้าที่ไหลเข้ามอตอร์สูงเกินกว่าที่วงจรควบคุมมอเตอร์จะทำได้ รวมทั้งป้องกันกรณีเกิดการลัดวงจรด้วย

วงจรป้องกันจะทำงานเมื่อมีกระแสไฟฟ้าไหลในช่วง 18 ถึง 36A

ระบบป้องกันแรงดันต่ำเกิน

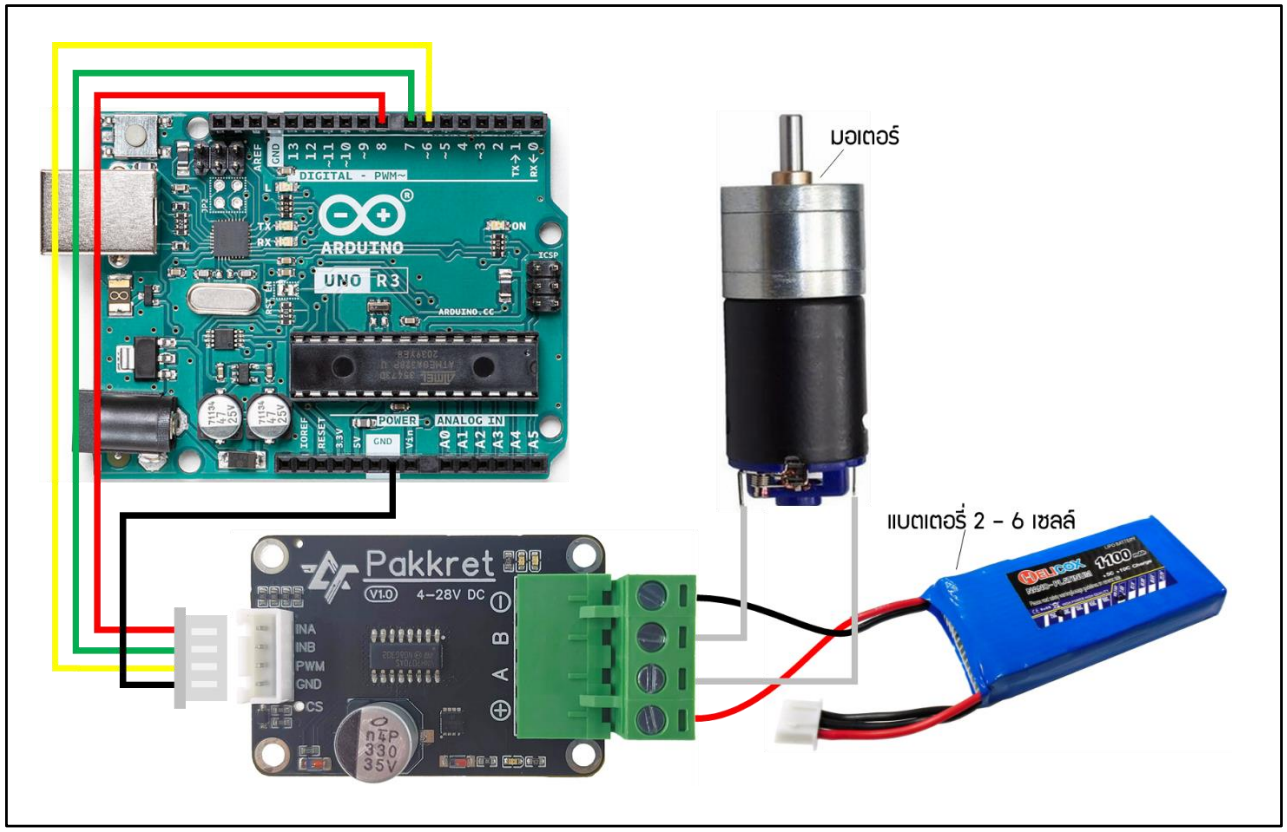
แรงดันอินพุตยิ่งต่ำ ยิ่งทำให้ประสิทธิ์ภายการนำกระแสไฟฟ้าของวงจรควบคุมมอเตอร์ต่ำลง หากแรงดันต่ำมากเกินไปอาจทำให้ประสิทธิ์ภาพการนำกระแสต่ำลงมากจนในที่สุดพลังงานเกือบทั้งหมดจะถูกสะสมภายในชิปและระบายผ่านความร้อนจนไปทำลายโครงสร้างของซิลิคอนภายในชิปขับมอเตอร์ ดังนั้นวงจรป้องกันแรงดันต่ำเกินจะช่วยป้องกันไม่ให้แรงดันไฟฟ้าที่ต่ำมากเกินไปส่งผลให้อาจจะเกิดความเสียหายได้

ระบบป้องกันจะทำงานเมื่อมีไฟเลี้ยงต่ำกว่า 4V และจะกลับมาทำงานได้อีกครั้งเมื่อมีแรงดันไฟเสียงสูงกว่า 5V

การใช้งานร่วมกับบอร์ด Arduino

บอร์ด Arduino Uno R3 มีขา PWM ทั้งหมด 6 ขา ประกอบด้วยขา 3, 5, 6, 9, 10, 11 โดยสามารถใช้งานได้ทั้งโหมด INA, INB, PWM และโหมด INA, INB

การต่อวงจรใช้งานในโหมด INA, INB, PWM แสดงดังรูปที่ 2



รูปที่ 2 การต่อบอร์ดขับมอเตอร์ปากเกร็ดเข้ากับบอร์ด Arduino แบบใช้ขา INA, INB, PWM, GND

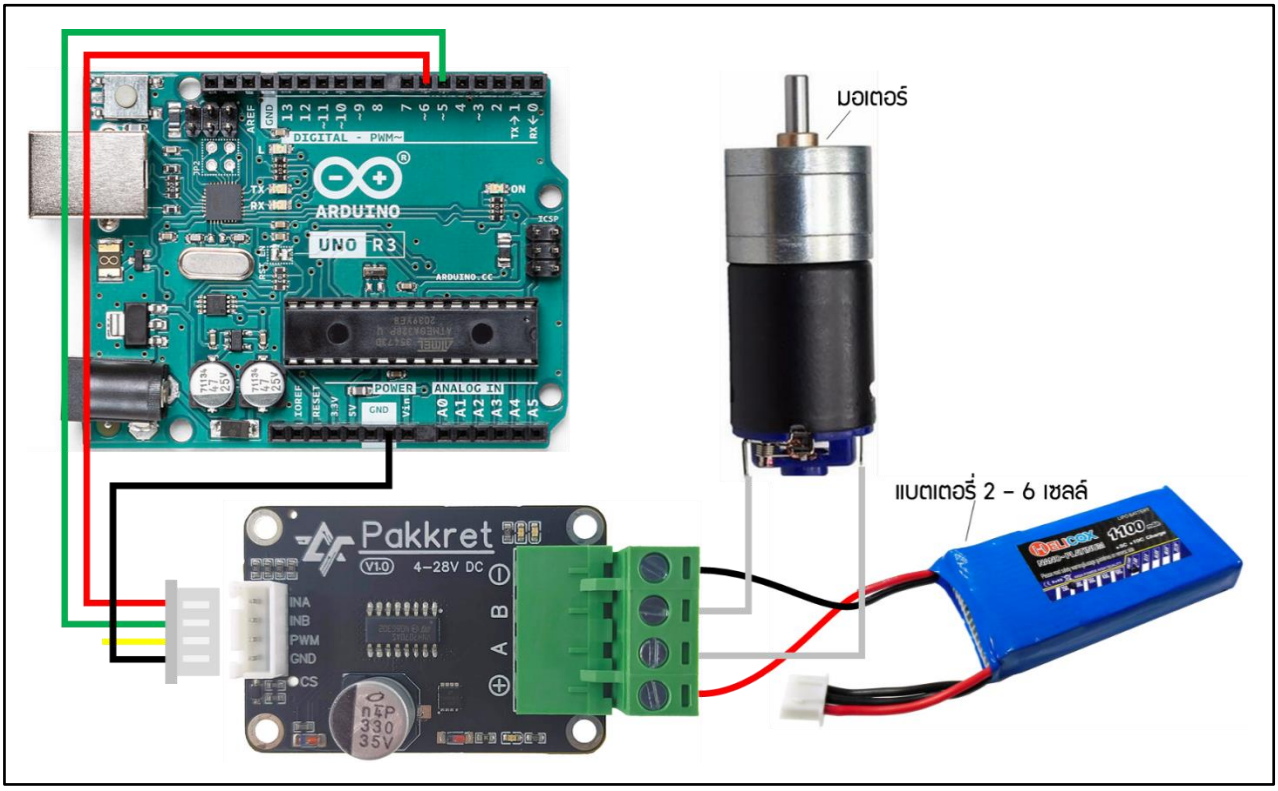
การเขียนโปรแกรมสั่งงานด้วย Arduino IDE ใช้คำสั่ง digitalWrite() เพื่อกำหนดทิศทางการหมุนผ่านการป้อนลอจิกที่ขา INA, INB และใช้คำสั่ง analogWrite() เพื่อกำหนดความเร็วการหมุน โดย 0 = 0%, 255 = 100%

ARTRONSHOP
www.artronshop.co.th

```
#define IN1_PIN 8 // กำหนดขาต่อ IN1 เป็นขา 8
#define IN2_PIN 7 // กำหนดขาต่อ IN2 เป็นขา 7
#define PWM_PIN 6 // กำหนดขาต่อ PWM เป็นขา 6

void setup() {
  pinMode(IN1_PIN, OUTPUT); // กำหนดขาที่ IN1 เป็น OUTPUT
  pinMode(IN2_PIN, OUTPUT); // กำหนดขาที่ IN2 เป็น OUTPUT
  pinMode(PWM_PIN, OUTPUT); // กำหนดขาที่ PWM เป็น OUTPUT
}

void loop() {
  // กำหนดให้มอเตอร์หมุนตามเข็ม
  digitalWrite(IN1_PIN, 1); // กำหนดลอจิก IN1 = 1
  digitalWrite(IN2_PIN, 0); // กำหนดลอจิก IN2 = 0
  analogWrite(PWM_PIN, 127); // กำหนดความเร็ว 50% (50% = 127)
  delay(1000); // ให้ค้างการทำงาน 1 วินาที
  analogWrite(PWM_PIN, 255); // กำหนดความเร็ว 100% (100% = 255)
  delay(1000); // ให้ค้างการทำงาน 1 วินาที

  // กำหนดให้มอเตอร์หมุนทวนข็ม
  digitalWrite(IN1_PIN, 0); // กำหนดลอจิก IN1 = 0
  digitalWrite(IN2_PIN, 1); // กำหนดลอจิก IN2 = 1
  analogWrite(PWM_PIN, 127); // กำหนดความเร็ว 50%
  delay(1000); // ให้ค้างการทำงาน 1 วินาที
  analogWrite(PWM_PIN, 255); // กำหนดความเร็ว 100%
  delay(1000); // ให้ค้างการทำงาน 1 วินาที

  // กำหนดมอเตอร์หยุดหมุน
  analogWrite(PWM_PIN, 0); // กำหนดความเร็ว 0%
  delay(1000); // ให้ค้างการทำงาน 1 วินาที
}
```

ดาวน์โหลดโค้ดโปรแกรมฉบับเต็ม คลิกที่นี่

การต่อวงจรใช้งานในโหมด INA, INB แสดงดังรูปที่ 3



รูปที่ 3 การต่อบอร์ดขับมอเตอร์ปากเกร็ดเข้ากับบอร์ด Arduino แบบใช้ขา INA, INB, GND

การเขียนโปรแกรมสั่งงานด้วย Arduino IDE ใช้คำสั่ง analogWrite() สั่งงานขา INA, INB โดย กำหนด 255 = ลอจิก 1, กำหนด 0 = ลอจิก 0, 1 – 254 = ความเร็วมอเตอร์ เมื่อ 1 มอเตอร์หมุนเร็ว และ 254 มอเตอร์หมุนช้า

ARTRONSHOP
www.artronshop.co.th

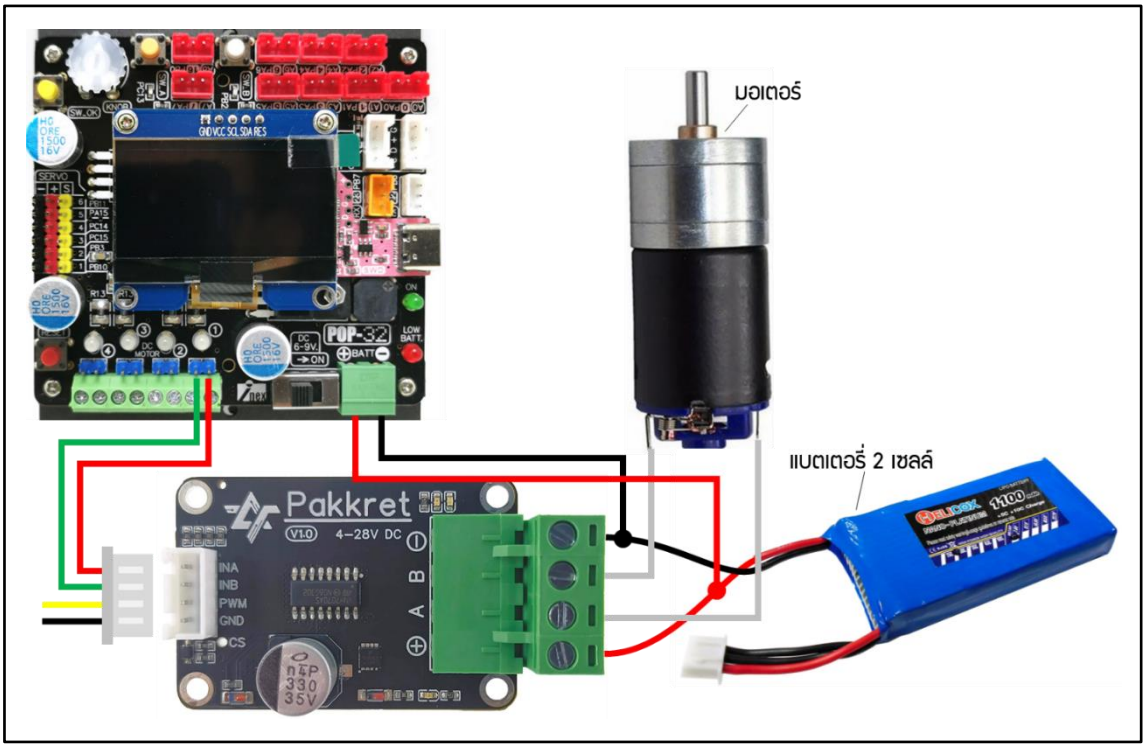
```
#define IN1_PIN 6 // กำหนดขาต่อ IN1 เป็นขา 6
#define IN2_PIN 5 // กำหนดขาต่อ IN2 เป็นขา 5

void setup() {
  pinMode(IN1_PIN, OUTPUT); // กำหนดขาที่ IN1 เป็น OUTPUT
  pinMode(IN2_PIN, OUTPUT); // กำหนดขาที่ IN2 เป็น OUTPUT
}

void loop() {
  // กำหนดให้มอเตอร์หมุนตามเข็ม
  analogWrite(IN1_PIN, 255); // ล็อกลอจิก IN1 = 1
  analogWrite(IN2_PIN, 127); // กำหนดความเร็ว 50% (50% = 127)
  delay(1000); // ให้ค้างการทำงาน 1 วินาที
  analogWrite(IN2_PIN, 0); // กำหนดความเร็ว 100% (100% = 0)
  delay(1000); // ให้ค้างการทำงาน 1 วินาที

  // กำหนดให้มอเตอร์หมุนทวนข็ม
  analogWrite(IN1_PIN, 127); // กำหนดความเร็ว 50% (50% = 127)
  analogWrite(IN2_PIN, 255); // ล็อกลอจิก IN2 = 1
  delay(1000); // ให้ค้างการทำงาน 1 วินาที
  analogWrite(IN1_PIN, 0); // กำหนดความเร็ว 100% (100% = 0)
  delay(1000); // ให้ค้างการทำงาน 1 วินาที

  // กำหนดมอเตอร์หยุดหมุน
  analogWrite(IN1_PIN, 255); // ล็อกลอจิก IN1 = 1
  analogWrite(IN2_PIN, 255); // ล็อกลอจิก IN2 = 1
  delay(1000); // ให้ค้างการทำงาน 1 วินาที
}
```

ดาวน์โหลดโค้ดโปรแกรมฉบับเต็ม คลิกที่นี่

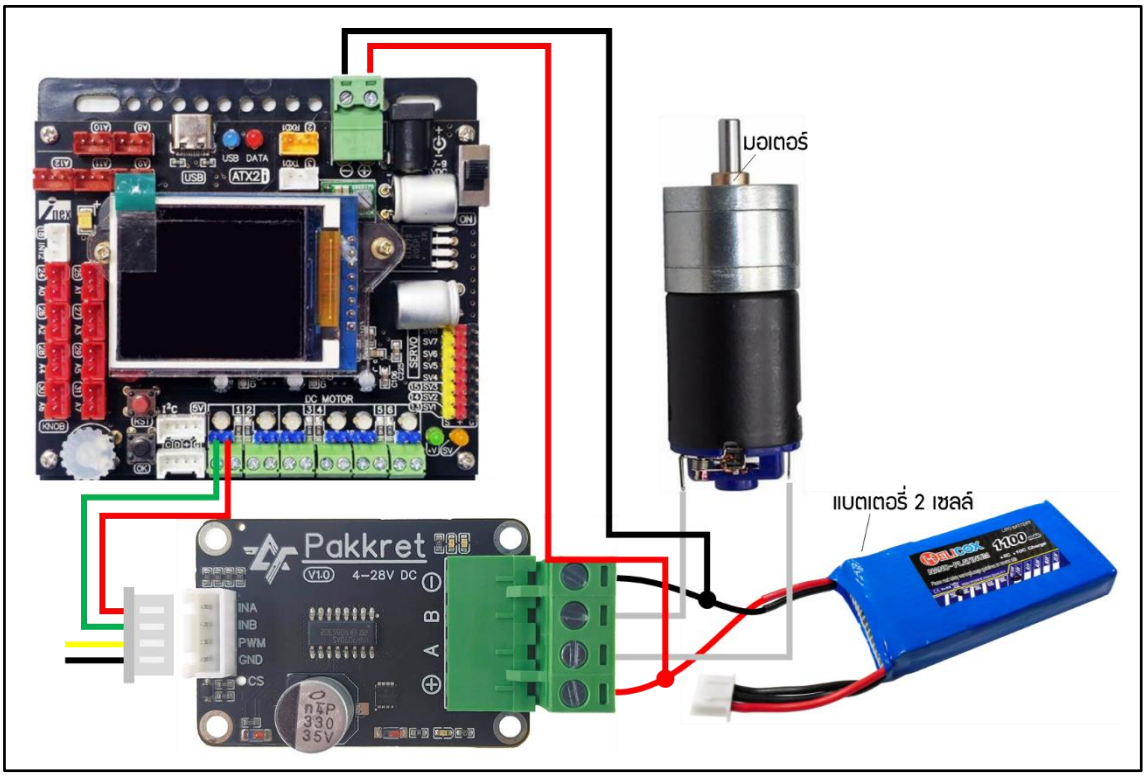
การใช้ขยายกำลังขับมอเตอร์ของบอร์ดควบคุมหุ่นยนต์

ด้วยข้อจำกัดด้านการออกแบบและความเหมาะสมในการใช้งาน บอร์ดควบคุมหุ่นยนต์ส่วนใหญ่จึงออกแบบให้วงจรขับมอเตอร์มีกำลังขับที่น้อย ถูกจำกัดด้วยกระแสไฟฟ้าและแรงดันไฟฟ้าขับมอเตอร์ที่ไม่มาก การเพิ่มกำลังขับสามารถใช้บอร์ดขับมอเตอร์ปากเกร็ดต่อขยายได้ ทำให้สามารถขับมอเตอร์ที่มีกำลังสูง ๆ ได้

การต่อวงจรเพื่อขยายกำลังขับสามารถต่อ INA, INB เข้าที่จุดต่อมอเตอร์ของบอร์ดควบคุมหุ่นยนต์ได้เลย การเขียนโปรแกรมควบคุมใช้โค้ดเดียวกันจากผู้ผลิตบอร์ดได้เลย ตัวอย่างการใช้บอร์ดปากเกร็ดขยายกำลังขับมอเตอร์ของบอร์ด POP-32, ATX-2i/ATX2+ ใช้แบตเตอรี่ 2 เซลล์ เป็นแหล่งพลังงาน แสดงดังรูปที่ 4 - 5

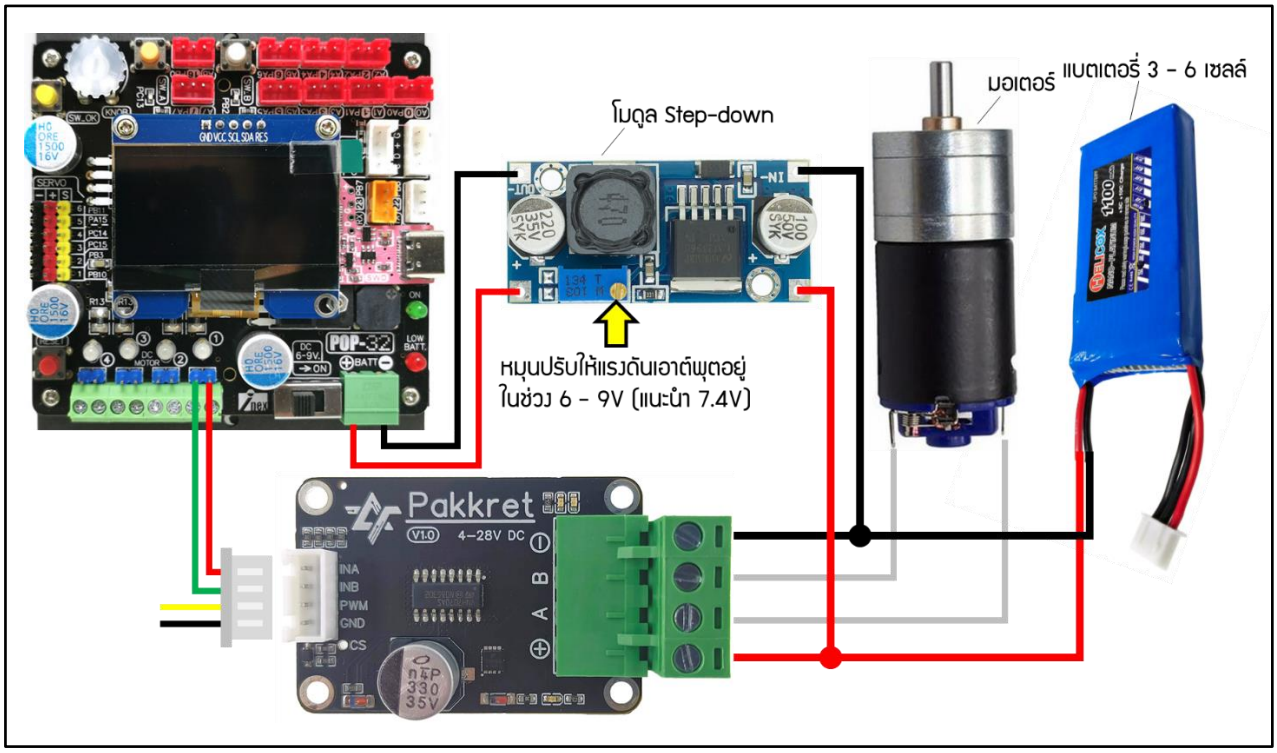


รูปที่ 4 การต่อบอร์ดขับมอเตอร์ปากเกร็ดเข้ากับบอร์ด POP-32 / POP-32i



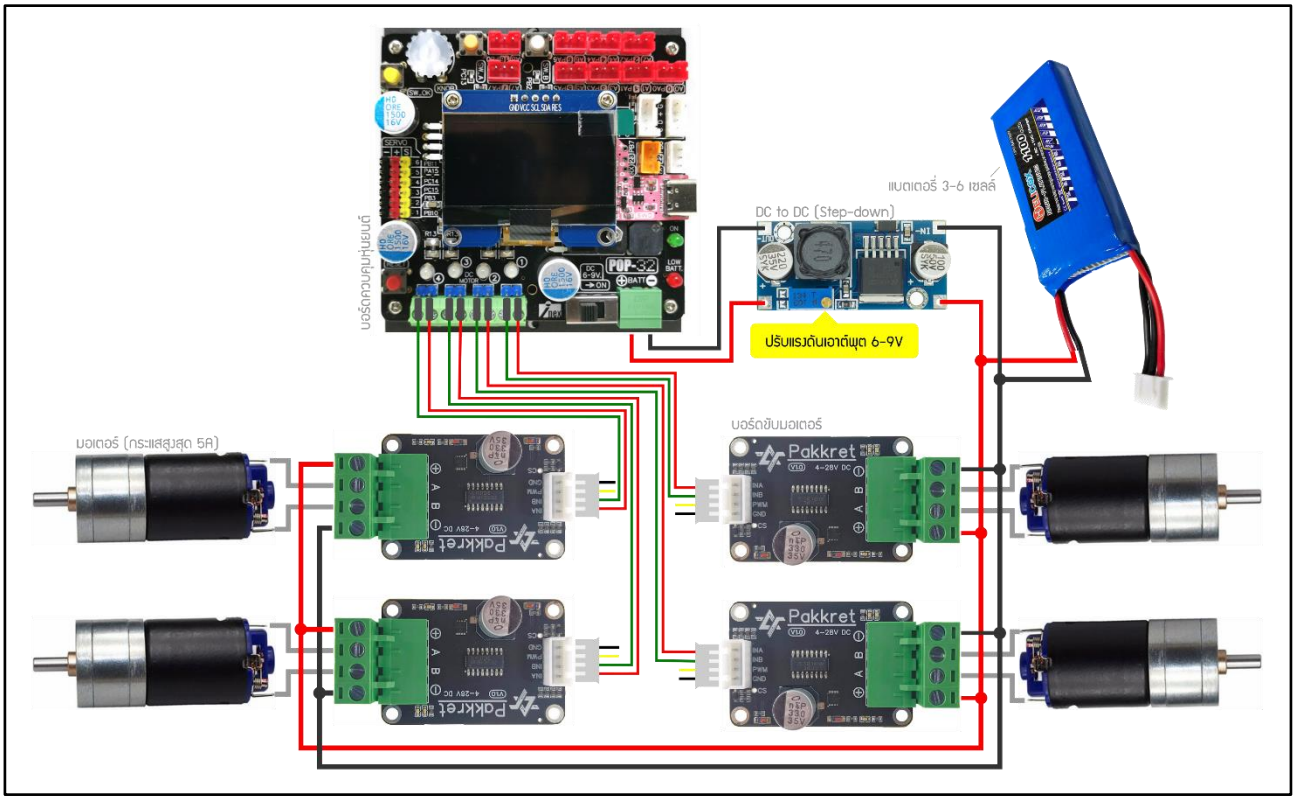
รูปที่ 5 การต่อบอร์ดขับมอเตอร์ปากเกร็ดเข้ากับบอร์ด ATX-2i / ATX2+

กรณีบอร์ดควบคุมหุ่นยนต์รองรับแรงดันไฟเลี้ยงไม่เกิน 9V (เช่น บอร์ด Arduino, POP-32, ATX-2i/ATX2+) แต่ต้องการขับมอเตอร์ 12V / 24V ให้ใช้โมดูล Step-down ลดแรงดันของแบตเตอรี่ให้ไม่เกิน 9V ต่อเข้ากับบอร์ดควบคุมหุ่นยนต์ แล้วต่อไฟจากแบตเตอรี่เข้าบอร์ดขับมอเตอร์ปากเกร็ดได้เลย ตัวอย่างการใช้บอร์ด POP-32i ขับมอเตอร์ 12V โดยใช้บอร์ดขับมอเตอร์ปากเกร็ดขยายกำลังขับ แสดงดังรูปที่ 6

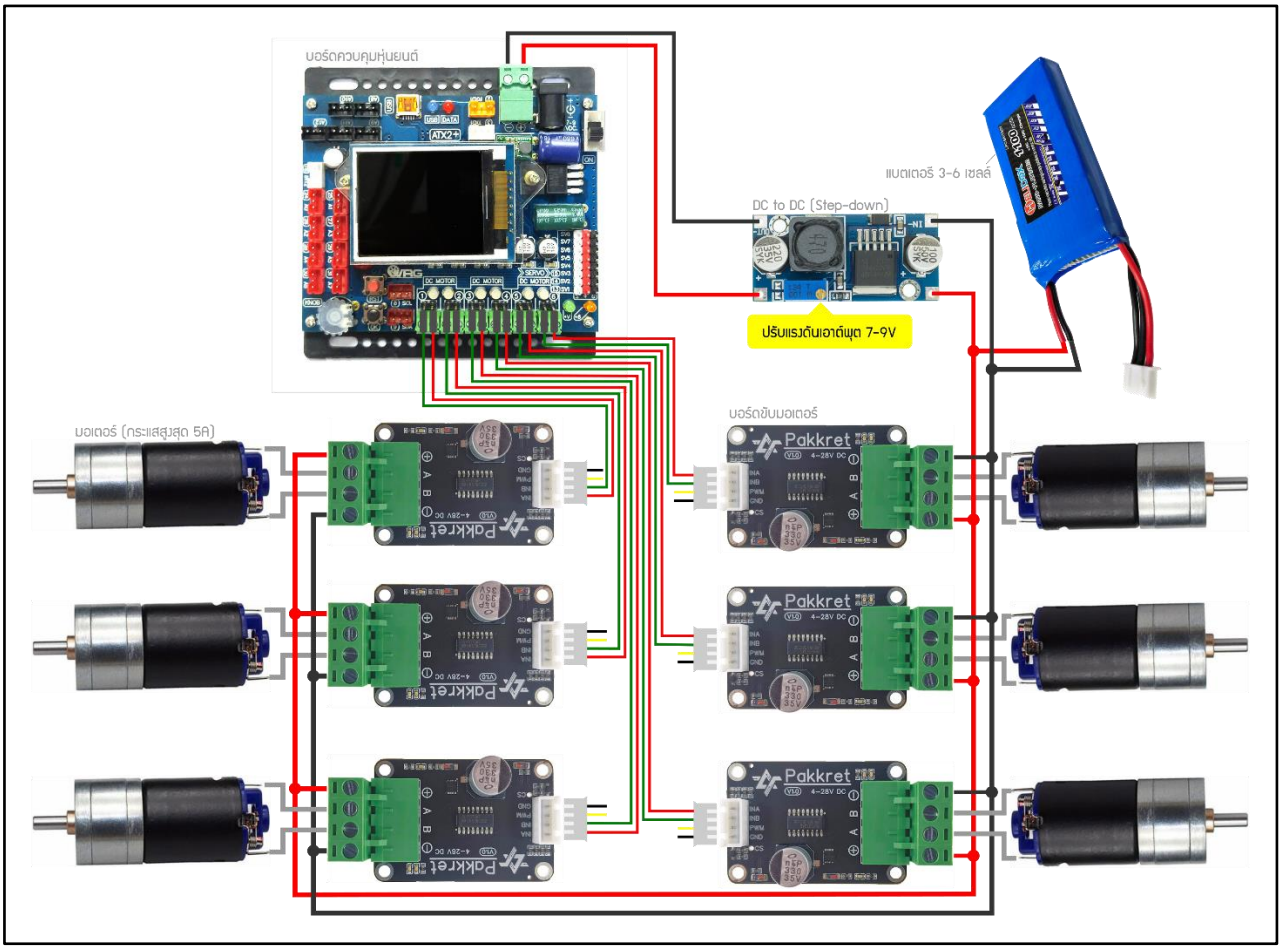


รูปที่ 6 การใช้บอร์ด POP-32i ขับมอเตอร์ 12V โดยใช้บอร์ดขับมอเตอร์ปากเกร็ดขยายกำลังขับ

การต่อใช้งานครบ 4-6 มอเตอร์ร่วมกับบอร์ด POP-32/ATX2+ แสดงดังรูปที่ 7-8



รูปที่ 7 การใช้บอร์ด POP-32i ขับมอเตอร์ 4 ช่อง โดยใช้บอร์ดขับมอเตอร์ปากเกร็ดขยายกำลังขับ



รูปที่ 8 การใช้บอร์ด ATX2+ ขับมอเตอร์ 4 ช่อง โดยใช้บอร์ดขับมอเตอร์ปากเกร็ดขยายกำลังขับ

ตัวอย่างการเขียนโค้ดโปรแกรมสั่งงานมอเตอร์โดยใช้คำสั่งของบอร์ด POP-32 / ATX2 แสดงด้านล่าง

```
#include <POP32.h>
void setup()
{

}
void loop()
{
  motor(1,30);
  motor(2,30);
  delay(3000);
  motor(1,-30);
  motor(2,-30);
  delay(3000);
}
```

ดาวน์โหลดโค้ดโปรแกรมฉบับเต็ม คลิกที่นี่

Soft Start

มอเตอร์บางรุ่นที่ใช้กระแสไฟฟ้าเริ่มหมุนสูงมาก อาจทำให้วงจรป้องกันกระแสเกินทำงาน แล้วทำให้มอเตอร์ไม่หมุนต่อ แนวทางแก้ไขคือทำ Soft Start โดยปล่อยสัญญาณ PWM จาก 0% แล้วค่อย ๆ เพิ่มไปจนถึงความเร็วที่ต้องการ

ตัวอย่างต้องการให้มอเตอร์หมุนที่ความเร็ว 0 ถึง 50% โดยทำ Soft Start ตอนเริ่มต้นหมุน เขียนโค้ดโปรแกรมได้ดังนี้

```
#define IN1_PIN 6 // กำหนดขาต่อ IN1 เป็นขา 6
#define IN2_PIN 5 // กำหนดขาต่อ IN2 เป็นขา 5

void soft_start(int pin, int start, int end) {
  if (start < end) {
    for (int i=start;i<=end;i++) {
      analogWrite(pin, i);
      delay(5);
    }
  } else if (start > end) {
    for (int i=start;i>=end;i--) {
      analogWrite(pin, i);
      delay(5);
    }
  }
}

void setup() {
  pinMode(IN1_PIN, OUTPUT); // กำหนดขาที่ IN1 เป็น OUTPUT
  pinMode(IN2_PIN, OUTPUT); // กำหนดขาที่ IN2 เป็น OUTPUT
}

void loop() {
  // กำหนดให้มอเตอร์หมุนตามเข็ม
  analogWrite(IN1_PIN, 255); // ล็อกลอจิก IN1 = 1
  soft_start(IN2_PIN, 255, 127); // กำหนดความเร็ว 0% -> 50%
  delay(3000); // ให้ค้างการทำงาน 1 วินาที
  soft_start(IN2_PIN, 127, 0); // กำหนดความเร็ว 50% -> 100%
  delay(3000); // ให้ค้างการทำงาน 1 วินาที
  soft_start(IN2_PIN, 0, 255); // กำหนดความเร็ว 100% -> 0%
  delay(3000); // ให้ค้างการทำงาน 1 วินาที
}
```

ดาวน์โหลดโค้ดโปรแกรมฉบับเต็ม คลิกที่นี่

ข้อแนะนำการใช้งาน

สายไฟที่เลือกใช้

- สายไฟที่ใช้ต่อจากแบตเตอรี่ และมอเตอร์ต้องมีขนาดใหญ่พอที่จะรองรับกระแสชั่วขณะของมอเตอร์ได้ โดยหากสายไฟมีขนาดใหญ่ไม่พออาจจะทำให้สายไฟร้อนขณะใช้งานจนเกิดการละลายของฉนวนหุ้มสายไฟได้ นอกจากนี้สายไฟที่เล็กจะไปต้านการไหลของกระแสไฟฟ้า ทำให้กระแสไฟฟ้าไม่สามารถไหลเข้าบอร์ดขับมอเตอร์/มอเตอร์ได้พอ
- สายไฟทองแดงควรมีการบัดกรีเคลือบสายเพื่อป้องกันสนิม (ออกไซด์) หากไม่มีการบัดกรีเคลือบแล้วเกิดออกไซด์จะทำให้กระแสไฟฟ้าไหลผ่านสายไฟได้ยาก ส่งผลให้เกิดความร้อนบริเวณออกไซด์ จนอาจจะเกิดไฟลุกไหม้ได้ในที่สุด

แหล่งจ่ายที่เลือกใช้

- ควรเลือกใช้แบตเตอรี่ที่มีความต้านทานภายในต่ำ โดยยิ่งต่ำยิ่งหมายถึงการจ่ายกระแสไฟฟ้าแบบต่อเนื่องได้มาก หากแบตเตอรี่ไม่สามารถจ่ายไฟได้เพียงพอจะส่งผลให้ไฟเลี้ยงตก จนวงจรป้องกันแรงดันต่ำทำงานได้
- ควรเลือกใช้แหล่งจ่าย/แบตเตอรี่ที่มีระบบป้องกันการลัดวงจร

มอเตอร์ที่เลือกใช้

- รองรับมอเตอร์เหลือง, N20, 16GA, GM20, GM25 และมอเตอร์อื่น ๆ ที่ใช้กระแสไฟฟ้าไม่เกิน 5A ต่อเนื่อง
- หากมอเตอร์ใช้กระแสไฟฟ้าต่อเนื่องเกิน 5A ระบบป้องกันกระแสเกินจะทำงาน
- กรณีมอเตอร์ใช้กระแสชั่วขณะสูงจนระบบป้องกันทำงาน ให้ทำ Soft start ด้วยซอฟแวร์เพื่อแก้ไขปัญหาดังกล่าว

มีคำถามเพิ่มเติมโปรดติดต่อสอบถามผ่าน LINE@ ได้โดยตรง

บริษัท อาร์ทรอน ชอป จำกัด รับพัฒนาสินค้าอิเล็กทรอนิกส์ บอร์ดอิเล็กทรอนิกส์ ด้านระบบ IoT พัฒนาเว็บไซต์ระบบ IoT ด้วย ReactJS / Next.js รับทำระบบหลังบ้านให้อุปกรณ์ IoT รับเขียนเฟิร์มแวร์อุปกรณ์ IoT ด้วย Arduino IDE / PlatformIO / Atmel Studio / MPLAB IDE / STM32CubeIDE / IAR / ESP-IDF / Arm Keil รับจัดหาชิ้นส่วนอิเล็กทรอนิกส์ ออกใบเสนอราคา และใบกำกับภาษีได้

37/146 ซ.โรงเรียนสวนกุหลาบนนทบุรี ถ.ติวานนท์ ตำบลปากเกร็ด อำเภอปากเกร็ด จังหวัดนนทบุรี 11120

ติดต่อสอบถามข้อมูลเพิ่มเติม

www.ArtronShop.co.th

063 639 9127 (สำนักงาน)

contact@ArtronShop.co.th

@artronshop

ArtronShop

ArtronShop

หน่วยธุรกิจอื่นของกลุ่มอาร์ทรอน

artron Innovative

We develop eco–friendly industry 4.0 innovation with digital transformation for sustainability – ArtronInnovative.co.th

ฝึก Code ฝึก Kit เรียนรู้การใช้ชีวิตไปกับการเขียนโค้ด – fb.me/ArtronAcademy