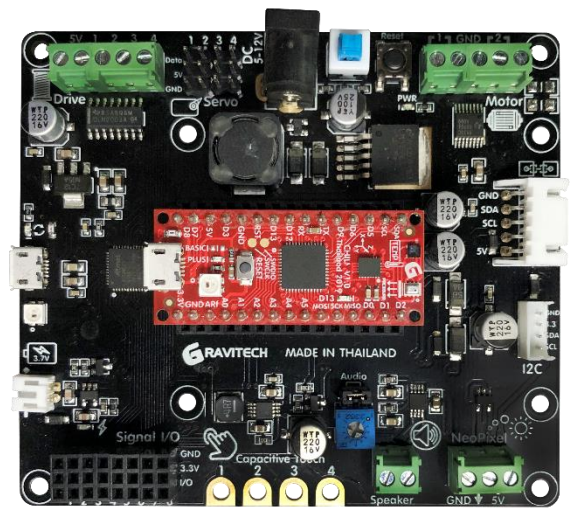
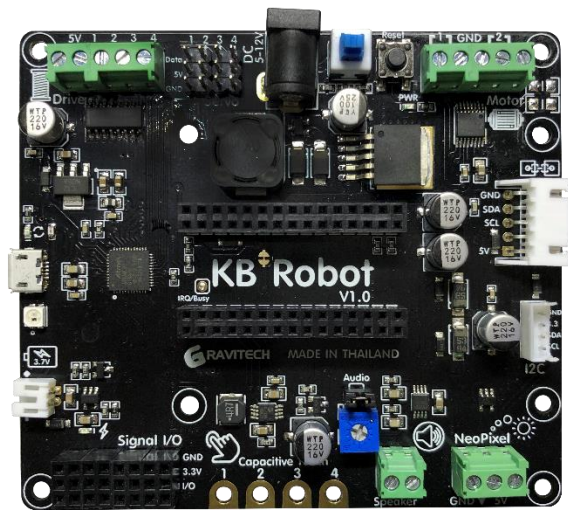
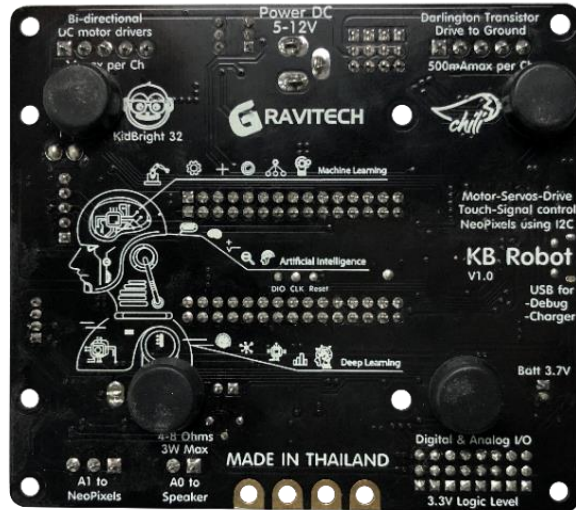


## KB ROBOT

## KB ROBOT



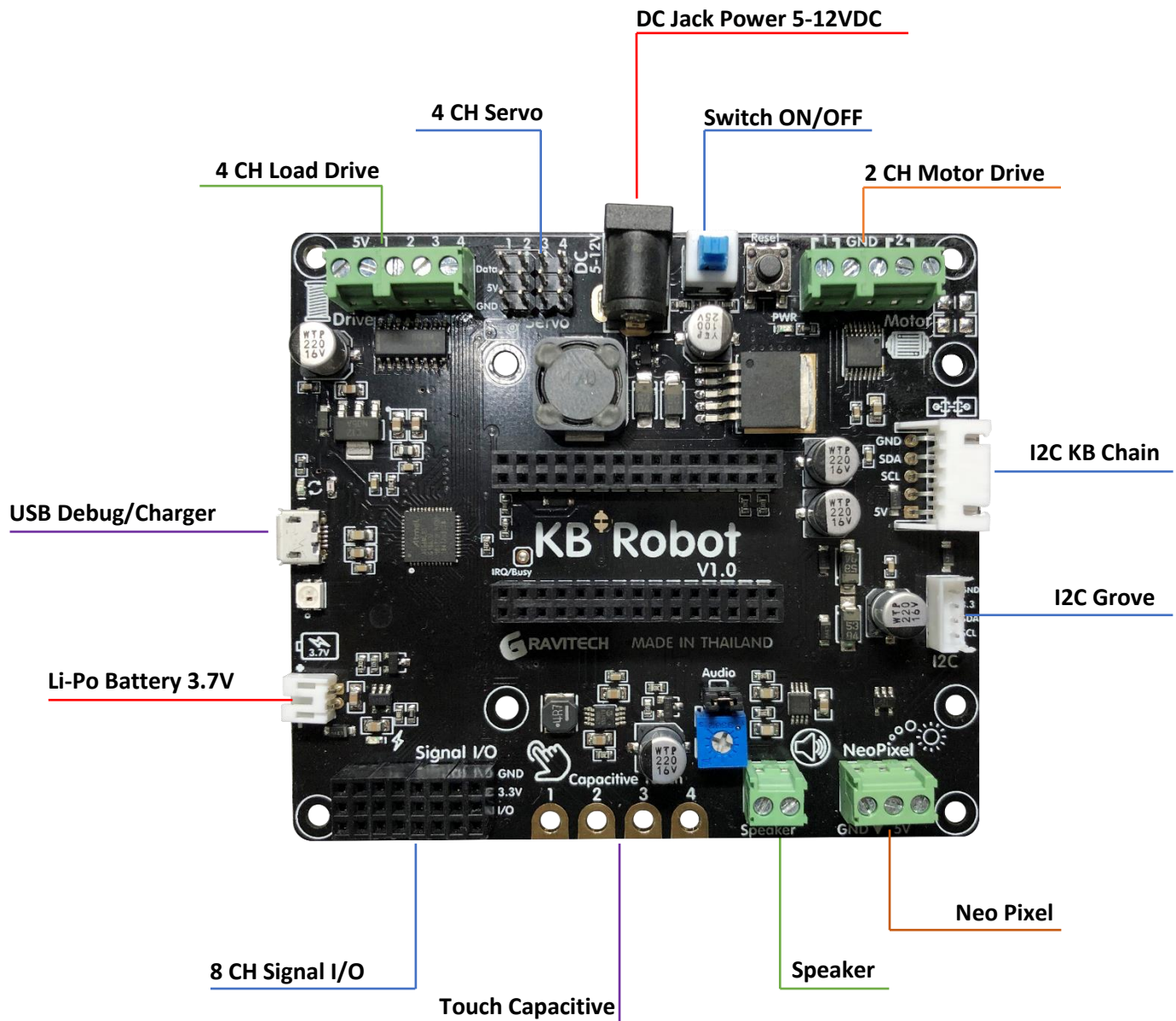
KB ROBOT คือบอร์ดเสริมที่ถูกออกแบบมาให้มีเซนเซอร์หรือฟังก์ชันการทำงานที่มากมาย เช่น ควบคุมมอเตอร์ ควบคุมลำโพง สามารถนำไปประยุกต์ใช้ได้หลากหลาย ไม่ว่าจะเป็นการประดิษฐ์หุ่นยนต์หรือรถบังคับ KB ROBOT นั้นถูกออกแบบมาให้รองรับการทำงานร่วมกับบอร์ด Arduino Nano Chili ได้และยังสามารถใช้งานร่วมกับ KidBright32 ได้อีกด้วย



## รายละเอียดของบอร์ด KB ROBOT

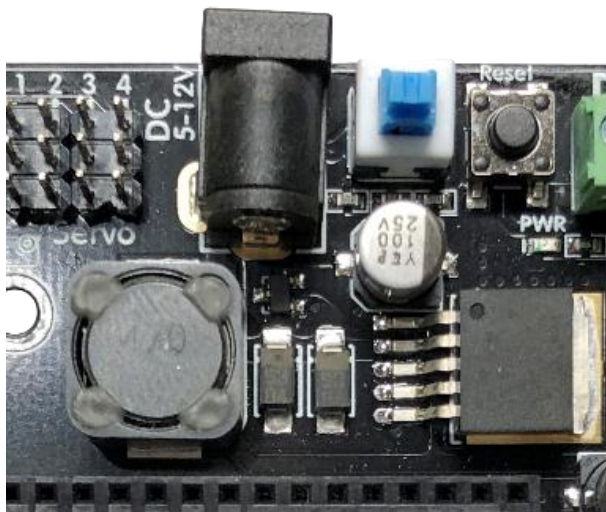
- 4 x Analog or Digital Servo control, with precision 16-bit timers
- 2 x Bi-directional brushed DC motor control, 1 Amp current limited each, with 8-bit PWM speed control (or one stepper)
- 4 x High current "Darlington" 500mA drive outputs with kick-back diode protection. For solenoids, relays, large LEDs, or one uni-polar stepper
- 4 x Capacitive touch sensors with alligator-pads
- 8 x Signal pins, digital in/out or analog inputs
- 1 x NeoPixel driver with 5V level shifter
- 1 x Class D, 4-8ohm speaker, 3W-max audio amplifier
- 1 x I2C Compatibility for KidBright and M5Stack

## ฟังก์ชันการทำงาน



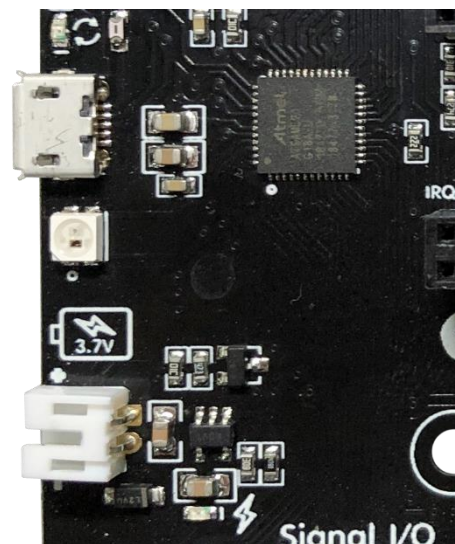


## แหล่งจ่ายไฟ



สามารถใช้ DC Jack Adapter ขนาด 2.1 มิลลิเมตร แรงดันไฟขนาด 5-12V เพื่อใช้ในการจ่ายไฟให้แก่ KB ROBOT ได้ โดยเมื่อจ่ายไฟแล้วตัว KB ROBOT จะทำการจ่ายไฟไปยัง NANO Chili โดยทันที

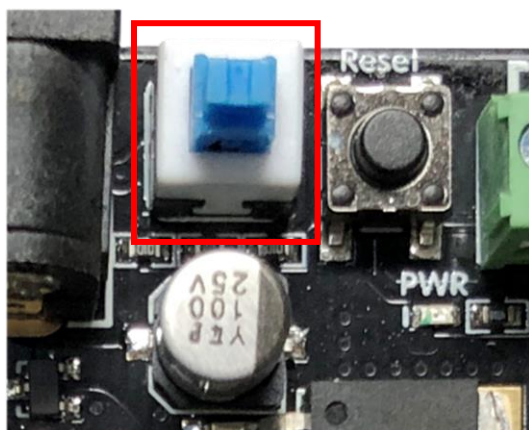
นอกจากการจ่ายไฟผ่านทางแหล่งจ่ายไฟ DC Jack โดยตรงแล้วตัว KB ROBOT ยังรองรับการจ่ายไฟแบบแบตเตอรี่อีกด้วยโดยสามารถนำแบตเตอรี่แบบ Li-Po แรงดัน 3.7V มาต่อได้และยังได้มีการใส่ชุดวงจรขาดแบตเตอรี่ไว้แล้วบนบอร์ดอีกด้วยโดยสามารถชาร์จแบตเตอรี่ผ่านทางสาย Micro USB โดยกระแสสูงสุดที่สามารถจ่ายได้อยู่ที่ 500mAh



\*แบตเตอรี่ที่แนะนำให้ใช้ Li-Po 3.7V 1150 - 1200mAh ใช้เวลาชาร์จประมาณ 3-4 hr.

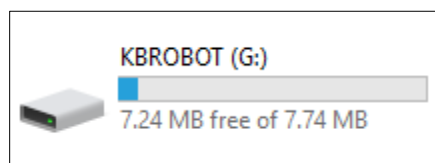
\*\*สามารถใส่แบตเตอรี่ที่มีจุมากกว่าหรือน้อยกว่าได้เช่นกันแต่ระยะเวลาในการชาร์จแบบจะขึ้นอยู่กับขนาดของแบตเตอรี่ที่ใช้งานด้วย

## สวิต เปิด/ปิด และ รีเซต



KB ROBOT นั้นมีปุ่มเปิดและปิด(1) การทำงานของบอร์ดโดยจะเป็นปุ่มกดติดกดดับ สามารถกดลงไปเพื่อให้บอร์ดทำการเริ่มการทำงาน และทำการกดอีกครั้งเพื่อปิดการทำงาน

KB ROBOT การกดปุ่ม Reset(2) นั้นไม่ใช่เพียงแค่กดเพื่อเริ่มต้นการทำงานใหม่ของบอร์ดเพียงอย่างเดียวแต่บอร์ด KB ROBOT นั้นสามารถเข้าสู่ Bootloader ได้ผ่านการกดปุ่ม Reset ได้ เพียงกดปุ่ม Reset ติดกัน 2 ครั้งจากนั้นจะขึ้นไดร์ที่มีชื่อว่า KBROBOT ขึ้นมา สามารถใช้ในการอัปเดตโปรแกรมเข้าไปควบคุมการทำงานได้



การกดปุ่ม Reset บนบอร์ด KB ROBOT นั้นไม่ได้เพียงแค่ทำการ Reset ตัวเอง แต่จะทำให้บอร์ด Arduino Nano Chili นั้นถูก Reset ไปด้วยหากไม่ต้องการให้การกดปุ่ม Reset นี้มีผลกับบอร์ด KB ROBOT สามารถทำการใช้คัตเตอร์ตัดลายวงจรบนบอร์ด KB ROBOT .ให้ขาดออกจากกันได้(3)



(1): ปุ่ม Power ON/OFF จะอยู่ในกรอบสีแดงสี่เหลี่ยม

(2): ปุ่ม Reset ในกรอบสีฟ้า

(3): เมื่อทำการตัดลายวงจรแล้วจะทำให้บอร์ด KB ROBOT ไม่สามารถกดปุ่ม Reset ได้อีก ต้องกดปุ่มเปิด ปิดเพื่อเริ่มการทำงานใหม่เท่านั้น

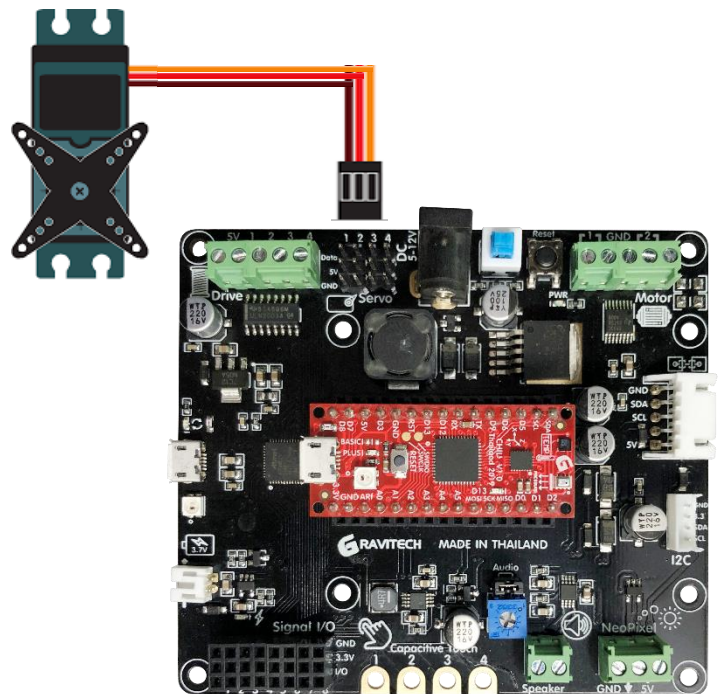
## Servo Driver Control



ในบอร์ด KB ROBOT นั้นมีความสามารถในการควบคุม Servo ได้ โดย Servo นั้นถือเป็นมอเตอร์ที่ควบคุมได้ที่สามารถหาซื้อได้ง่าย ราคาไม่สูงมาก และลง่ายนยในการใช้งานดังนั้น ทำให้มีผู้ใช้งานอยู่เป็นจำนวนไม่น้อยนัก

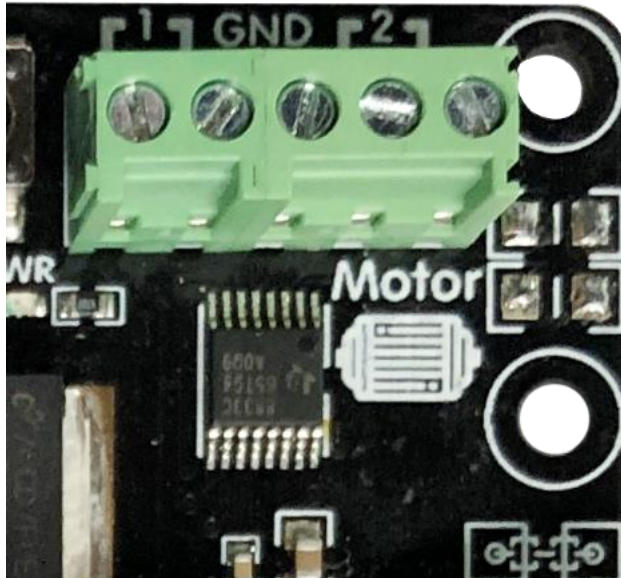
โดย KB ROBOT นั้นสามารถรองรับการควบคุม Servo พร้อมกันได้สูงสุดถึง 4 ตัว จึงสามารถนำไปประยุกต์สร้างเป็นขาของหุ่นยนต์เดินได้

สามารถใช้ Servo ขนาดใดก็ได้ในการรต่อเพื่อควบคุม Servo โดย Servo นั้นต้องเป็นแบบที่สามารถหมุนได้ 180 องศา และใช้สายในการเชื่อมต่อ 3 เส้น โดยมีไฟเลี้ยงที่ 3.3V

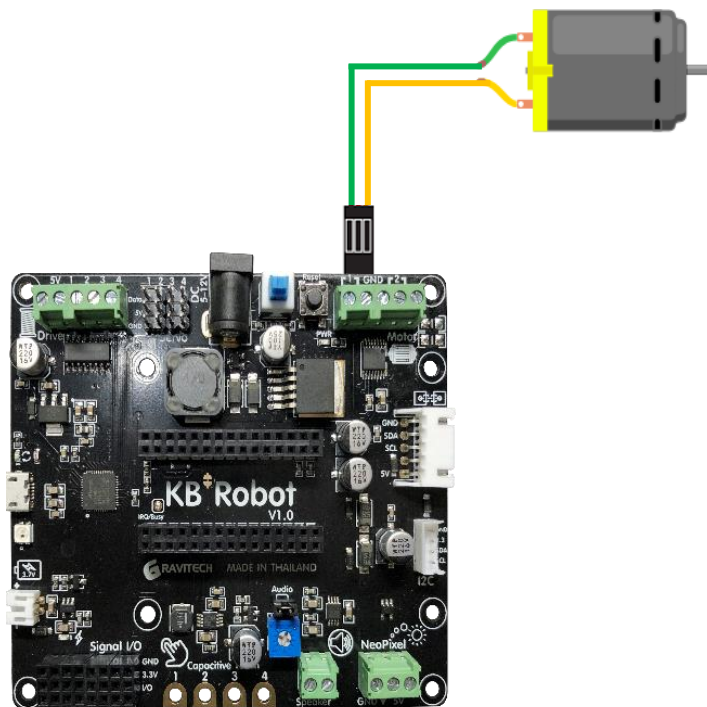


- ขาที่ใช้ในการควบคุมของ Servo นั้นต่อด้วยขาหมายเลข 14,15,16,17
- ขา PWM ของ Servo นั้นเป็นแบบ 16bit
- ไฟที่ KB ROBOT สามารถจ่ายไปเลี้ยง Servo ได้นั้นคือ 3.3V เท่านั้น

## Motor Drive 2CH



โดยปกติแล้ว Motor เมื่อจ่ายไฟแล้วมันจะทำการหมุนอย่างต่อเนื่องในทิศทางเดียวตลอดจนกว่าจะทำการตัดไฟ หากต้องการจะสลับทิศทางการทำงานนั้นจะต้องต่อไฟสลับขั้วกัน ทำให้เป็นปัญหาในการใช้งานในหลายๆด้าน ดังนั้น Motor Driver จึงเข้ามามีบทบาทในการควบคุม Motor เป็นอย่างมาก เนื่องจากต่อ Motor เพียงแค่ 2 สายเท่านั้นเราสามารถควบคุมได้ทั้งความเร็วในการหมุนและทิศทางการทำงานได้อย่างอิสระเลยทีเดียว

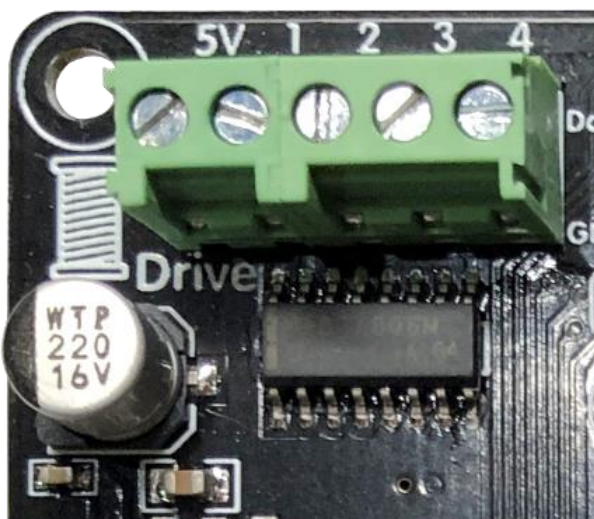


KB ROBOT นั้นมีการออกแบบให้มีวงจรขับ Motor ถึง 2 channel หมายความว่าสามารถควบคุม Motor ได้พร้อมกันถึง 2 ตัว โดย Motor ที่สามารถควบคุมได้มีแรงดันอยู่ที่ 3 – 6V

(1): ขาที่ใช้ในการควบคุมของ Motor Driver นั้นต่อด้วยขาหมายเลข 22+23 (motor1) และ 19+18(motor2)



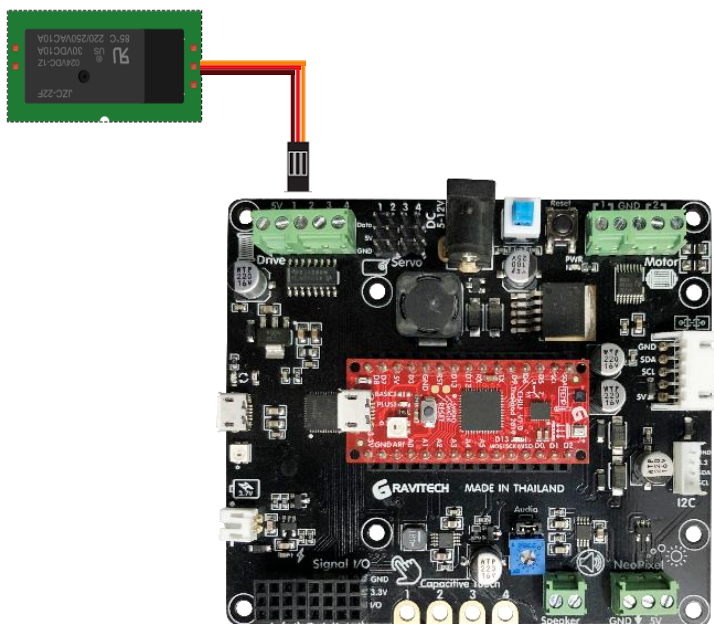
## High Power Drivers



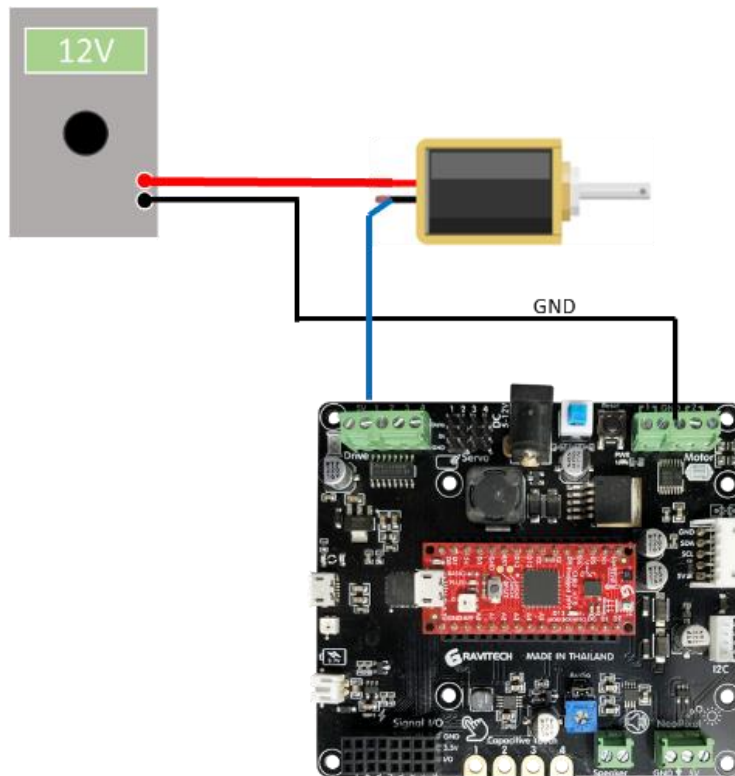
ในบางครั้งเราไม่ได้แค่ต้องการควบคุมแค่ Servo หรือ DC Motor เท่านั้นบ้างครั้งเราอาจจะต้องการที่จะควบคุมการทำงานอุปกรณ์ที่มีกำลังสูงๆ หรืออุปกรณ์ที่ใช้กระแสไฟค่อนข้างสูงในการทำงาน เช่น รีเลย์ มอเตอร์ขนาดใหญ่ หรือหลอดไฟ LED ขนาดใหญ่ เป็นต้น แต่มักจะเจอปัญหาที่ว่าขาของบอร์ดส่วนใหญ่จะขับกระแสได้ค่อนข้างต่ำทำให้ไม่สามารถใช้งานได้หรือถ้าอยากใช้งานจริงก็จำเป็นต้องต่อวงจรขับเองซึ่งก็มีขนาดและความยุ่งยากอยู่พอสมควร

KB ROBOT นั้นได้แก้ปัญหาในข้อนี้ โดยการ เพิ่มวงจรขับ High Power Drivers เพื่อทำการขับอุปกรณ์ที่ต้องการกระแสสูงๆในการทำงานได้

โดยสามารถต่ออุปกรณ์ที่ใช้กำลังสูงต่อเข้ากับหัวต่อเทอร์มินอลได้ 4 Chanel โดยขาของ Output นั้นเมื่อถูกใช้งานจะทำการต่อลง GND



ในกรณีที่ต้องการจะนำไปประยุกต์ใช้งานกับอุปกรณ์ที่มีแรงดันสูงกว่าปกติ เช่น โซลินอย 12V นั้น สามารถทำได้ โดยการจ่ายไฟจากภายนอกให้กับอุปกรณ์ โซลินอย แล้วทำการต่อขา GND เข้ากับขา OUTPUT ของบอร์ด KB ROBOT ได้ และ GND จากแหล่งจ่ายภายนอก 12V จะต้องต่อร่วมกับ GND ของบอร์ดด้วย(1)

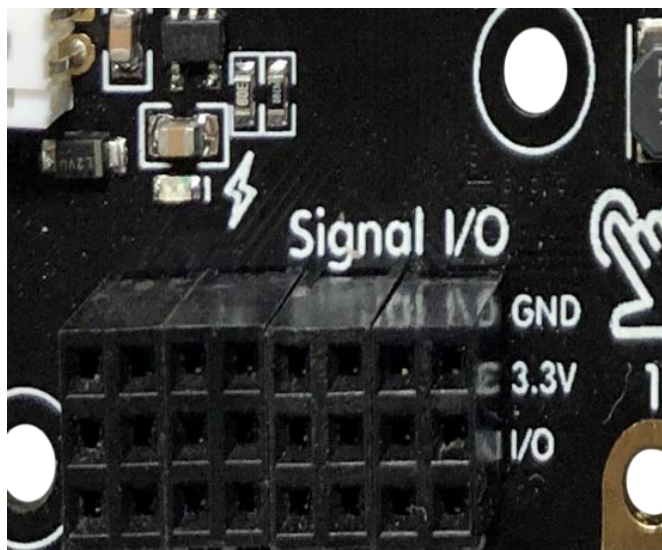


ตัวอย่างการต่อโซลินอยด์ที่ใช้แรงดัน 12V โดยใช้บอร์ด KB ROBOT ควบคุมการทำงาน

#### ข้อควรระวัง

- (1): การต่ออุปกรณ์ที่มีแรงดันไฟมากกว่า 5V สามารถทำได้แต่ต้องจ่ายจากภายนอกเท่านั้นและทำการเชื่อมต่อ GND ร่วมกับ บอร์ดด้วยเสมอ
- (2): กระแสที่สูงที่สุดที่สามารถรับได้อยู่ที่ 500mA
- (3): สามารถต่อไฟภายนอกได้สูงสุด 30VDC
- (4): ขาที่ใช้ในการควบคุม OUTPUT ขา 13, 12, 43, 42

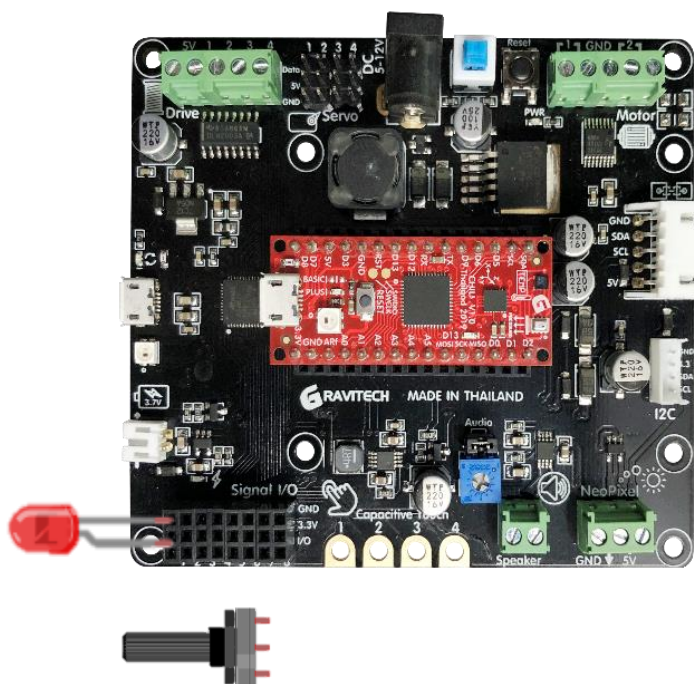
## Signal I/O 8CH



และแน่นอนว่าหากต้องการจะ blink LED หรือต้องการอ่านค่าที่ได้จาก Switch นั้นก็สามารถทำได้เช่นกันเพราะ KB ROBOT นั้นมีพ็อต Signal I/O มาเลือกใช้งานถึง 8 channel 8 คุณามารถใช้ในการอ่านค่า digital หรือ อ่านค่าจาก analog ก็ได้ หรือแม้แต่ต้องการควบคุมสัญญาณ digital ได้เช่นกัน เนื่องจากการออกแบบให้ Connector เป็นแบบ 3x8 ซึ่งมีขาไฟ 3.3V และ GND มาด้วยแล้วจะสะดวกที่จะนำเซนเซอร์มาต่อใช้งานได้ทันที

### คุณสมบัติของขาสัญญาณ

- ขาทั้งหมดของ Signal เป็น Logic แบบ 3.3V
- ขาทั้งหมดสามารถค่าแบบ analog ด้วยความละเอียดแบบ 12bit ได้
- ขาทั้งหมดสามารถควบคุม Logic High(3.3V) และ Low(0V) ได้
- ขาทั้งหมดสามารถขับเคลื่อนได้สูงสุดอยู่ที่ 7mA ในกรณีที่กำหนดเป็น Output
- ขาทั้งหมดนั้นต่อตัวต้านทานภายในเอาต์แล้วที่ 50k $\Omega$  pull-up ในกรณีที่กำหนดเป็น Input
- Signal IO จะมีขาหมายเลข 2, 3, 40, 41, 11, 10, 9, 8 ตามลำดับ



## Capacitive Touch



Capacitive touch หรือ เซนเซอร์จับการสัมผัส นั้นมีความสามารถในการรับการแตะหรือสัมผัสที่มาจากนิ้วมือหรือการสัมผัสที่เกิดจากโลหะได้ โดยแถบเซนเซอร์จะมีลักษณะเป็นแถบทองแดงที่เคลือบไว้ด้วยหมึกที่มีความสามารถในการนำไฟฟ้า KB ROBOT นั้นสามารถตรวจจับสัญญาณที่มาจากการแตะที่เกิดขึ้นบนแถบทองแดง Capacitive touch ได้ทั้ง 4 channel ได้

โดยคุณสามารถใช้นิ้วมือสัมผัสโดยตรงหรือจะใช้วัสดุที่เปียกน้ำมาสัมผัสก็ได้เช่นกัน(1) ตัวแถบ Capacitive touch นั้นก็สามารถอ่านค่าได้เช่นเดียวกัน

(1): การใช้วัสดุที่เปียกน้ำมาสัมผัสควรใช้สายไฟปากจระเข้มาหนีบที่แถบสัมผัสก่อนแล้วจึงนำปลายสายอีกด้านไปสัมผัสวัสดุที่เปลี่ยนน้ำ

(2): Capacitive touch นั้นมีหมายเลข 4, 5, 6, 7



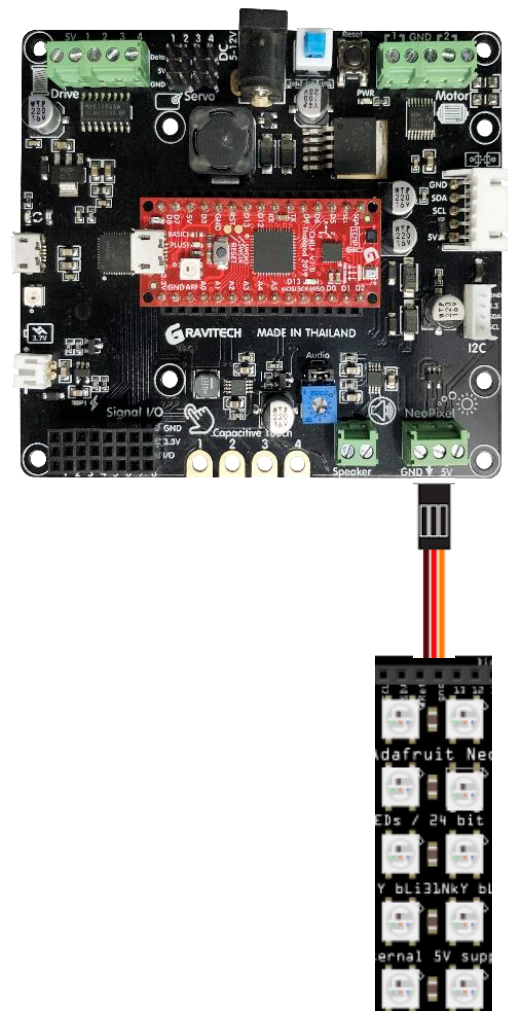
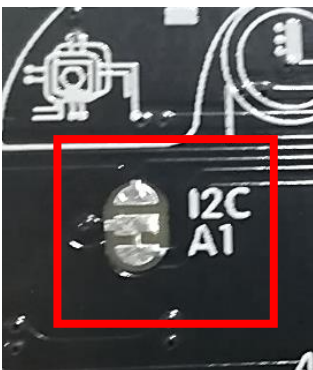
## NeoPixel Drive



แสงจากหลอด NeoPixel นั้นช่วยเพิ่มความสวยงามให้แก่หุ่นยนต์หรือโปรเจกต์ของคุณได้ มันง่ายมากหากคุณใช้ KB ROBOT ควบคุมหลอด NeoPixel เพราะ KB ROBOT นั้นมีหัวต่อเทอมินอลที่สามารถต่อเข้ากับ NeoPixel ได้โดยแบ่งออกเป็น GND Signal 5V รุ่นของ Chipset ที่รองรับคือ (WS2811/WS2811/SK6812) โดยขาสัญญานันมีวงจร Level Shifter 5V มาแล้ว แนะนำให้ใช้รุ่นที่ใช้ขาสัญญานที่ 5V เพื่อการใช้งานอย่างเต็มประสิทธิภาพ

KB ROBOT หากใช้งานร่วมกับ Arduino Nano Chili จะสามารถใช้งาน NeoPixel External ได้เลยโดยจะใช้ขา A1 (Pin15)

กรณีที่ไม่ได้ใช้งานร่วมกับ Arduino Nano Chili ก็สามารถใช้งานได้โดยการตัดลายวงจรที่ด้านหลังบอร์ด KB ROBOT แล้วทำการบัดกรี แผ่นบัดกรีขากลางเข้ากับขา I2C ก็จะสามารถใช้งาน NeoPixel ผ่าน I2C ได้



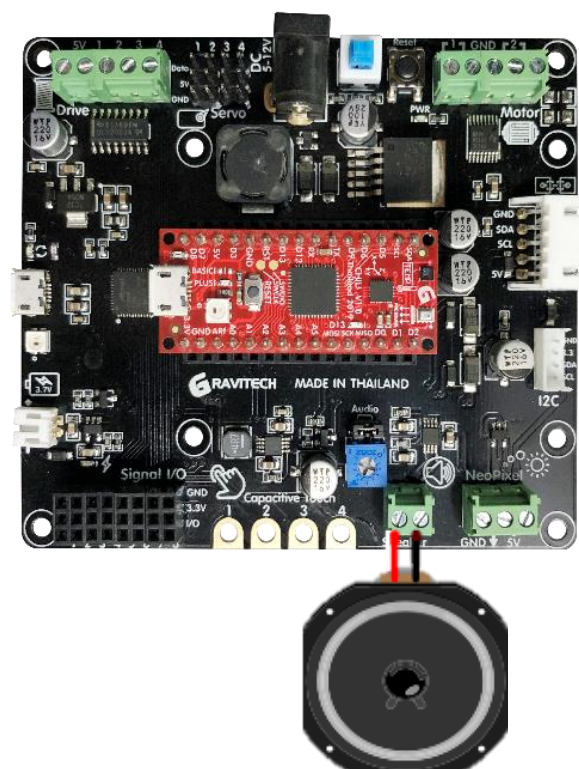
## Speaker Drive



คุณสามารถใช้ KB ROBOT ในการสร้างเสียงเพลงให้แก่หุ่นยนต์หรือแม้แต่โปรเจกของคุณได้ ใน KB ROBOT มีวงจรขยายเสียงให้แก่นักแล้ว แต่ต้องระลึกไว้เสมอว่า KB ROBOT ไม่สามารถส่งเสียงได้ด้วยตัวของมันเองเสียงที่ส่งออกมาจำเป็นต้องมาจากการควบคุมของ Arduino Nano Chili<sup>(1)</sup>

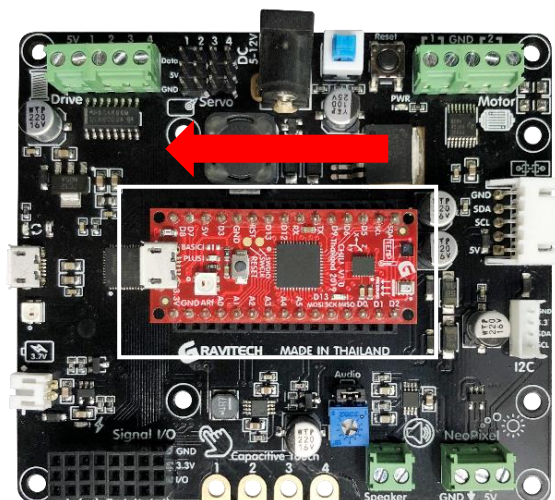
### คุณสมบัติของตัวขับลำโพง KB ROBOT

- เป็นตัวขยายเสียงแบบ Class D
- สามารถขยายลำโพงได้สูงสุดอยู่ที่  $4\Omega$  ถึง  $8\Omega$  โดยขับได้กำลัง 3W ที่  $4\Omega$  และ 1W ที่  $8\Omega$
- สามารถกำลังขยาย(ความดังของลำโพง) ได้จากการหมุนตัวต้านทานปรับค่าได้
- เทอมินอลที่ใช้ในการต่อลำโพงเป็นแบบ 5VDC ไม่ควรนำไปต่อและต่อมาจากวงจรขยายโดยตรง ไม่ควรนำไปต่อกับชุดลำโพงขนาดใหญ่หรือต่อไฟเข้าไปโดยตรง



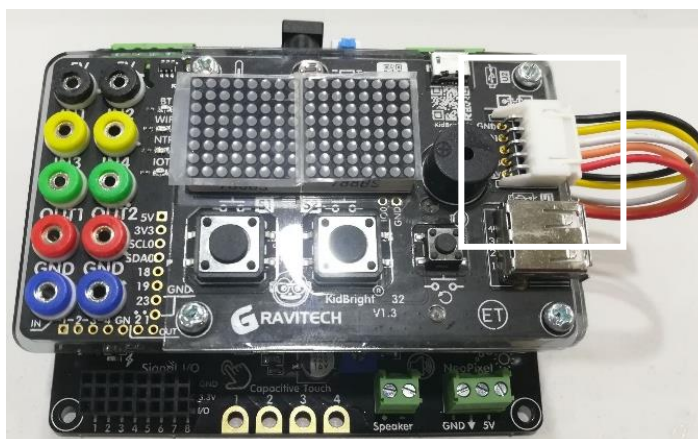
(1): สามารถใช้บอร์ดรุ่นอื่นควบคุมการทำงานของลำโพงได้เช่นเดียวกัน โดยการต่อขา PWM เข้ากับขา (A0, Pin14) บนตัวบอร์ด KB ROBOT

## รองรับหลายการเชื่อมต่อ

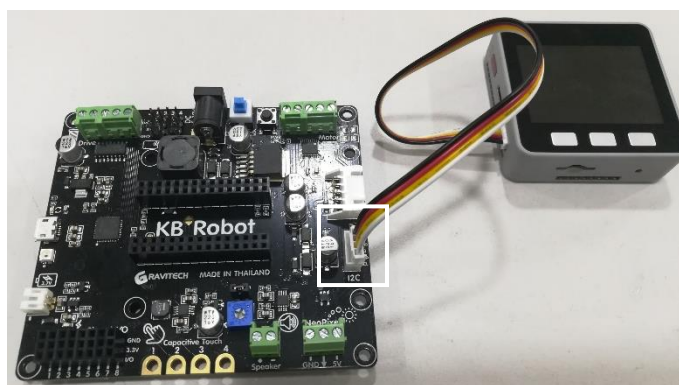


KB ROBOT ถูกออกแบบมาเพื่อรองรับการทำงานร่วมกับบอร์ด Arduino Nano Chili โดยเฉพาะ เนื่องจากถูกออกแบบมาสำหรับ Nano Chili แล้วดังนั้นจึงสามารถใช้ฟังก์ชันของบอร์ด KB ROBOT ได้อย่างครบถ้วนไม่ว่าจะเป็นการควบคุม Servo หรือแม้แต่การทำงานของลำโพงก็ตาม โดยให้หันทางด้าน Micro USB ไปตามแนวลูกศร(1)

KB ROBOT ไม่เพียงแต่จะรองรับแต่ Nano Chili บอร์ด KB ROBOT ยังสามารถรองรับการทำงานโดยตรงกับบอร์ด KidBright32 อีกด้วยโดยการเชื่อมต่อหลักจะเชื่อมต่อผ่านทางสาย KB Chain Connector ของทาง KidBright โดยตรงรูปแบบการสื่อสารจะอยู่ในรูปแบบของ I2C(2)



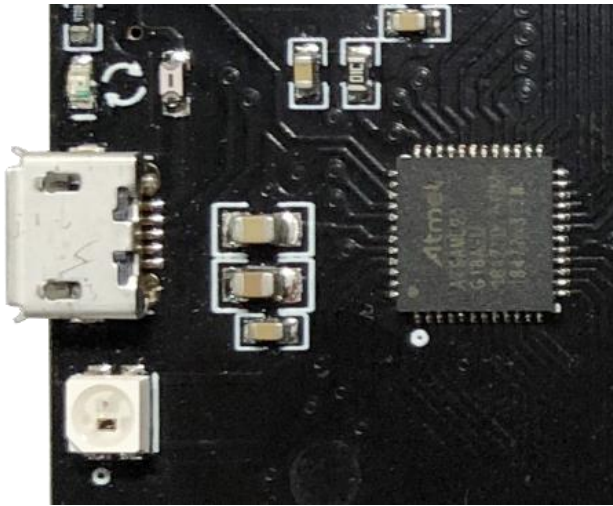
เช่นเดียวกับ M5Stack ก็สามารถใช้งานบอร์ด KB ROBOT เล่นกันผ่านทางสาย Grove ผ่านการสื่อสารแบบ I2C(3)



- (1): การต่อ Arduino Nano Chili ควรระวังขาที่เสียบลงบนบอร์ด KB ROBOT ควรตรวจสอบให้ดีก่อนจ่ายไฟมิเช่นนั้นอาจเกิดความเสียหายกับบอร์ดทั้ง 2 ได้
- (2): การต่อ KB ROBOT ผ่านสาย Chain Connector นั้นใน KidBright ใช้ I2C นะใช้หมายเลข 4,5
- (3): การต่อ KB ROBOT นสาย Grove แนะนำให้ใช้สายเฉพาะของ Grove Sensor เท่านั้น



## USB Programmer and Charger Battery



ใน KB ROBOT ใช้ ATSAMD21 เป็นหน่วยประมวลผลหลัก โดยจะมีหลอดไฟ NeoPixel ที่ใช้ในการบอกสถานะต่างๆ ของบอร์ดติดมาให้ เราสามารถควบคุมหลอดนี้ได้เช่นเดียวกัน(1)

หัวต่อ Micro USB นี้ใช้ในการอัปโหลดโปรแกรมและใช้ในการชาร์จแบตเตอรี่ให้ช่องต่อแบตเตอรี่เท่านั้น (2)ไม่สามารถใช้ในการจ่ายไฟเพื่อเลี้ยงทั้งบอร์ดให้ทำงานได้

### คุณสมบัติของ ATSAMD21

- ARM Cortex-M0+ CPU running at up to 48MHz 32bit MCU
- 256KB in-system self-programmable Flash
- 32KB SRAM Memory
- Power-on reset (POR) and brown-out detection (BOD)



(1): หลอด NeoPixel ภายในบอร์ดสามารถใช้หมายเลข 27 ในการควบคุมได้

(2): สามารถใช้ Micro USB ในการชาร์จแบตเตอรี่ได้สามารถดูข้อมูลแบตเตอรี่ได้ในหัวข้อ [แหล่งจ่ายไฟ](#)



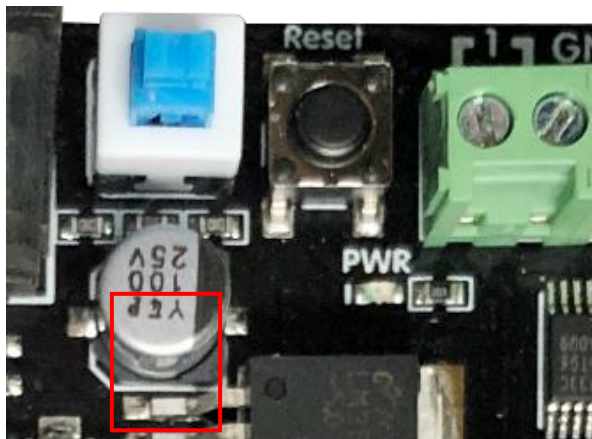
## เริ่มต้นการทำงานของ KB ROBOT

การใช้งาน KB ROBOT ก่อนที่เราจะเริ่มขงงานในฟังก์ชันต่างๆของ KB ROBOT ได้นั้นไม่ว่าจะเป็น Motor Drive, Servo Drive หรือ Signal I/O ได้นั้นเราจะต้องมีการใส่เฟิร์มแวร์ที่ใช้ในการควบคุมการทำงานให้แก่บอร์ด KB ROBOT ของเราก่อน

### ขั้นตอนการอัปเดตเฟิร์มแวร์

#### ขั้นตอนที่1: เริ่มต้นในการเสียบสาย Micro USB เข้ากับบอร์ด KB ROBOT

เริ่มต้นโดยการต่อสายที่เชื่อมต่อเข้ากับ PC/Laptop เข้ากับ KB ROBOT ของเราก่อนโดยเมื่อเชื่อมต่อแล้วเราจะเห็นไฟสถานะที่หลอด NeoPixel ติดขึ้น

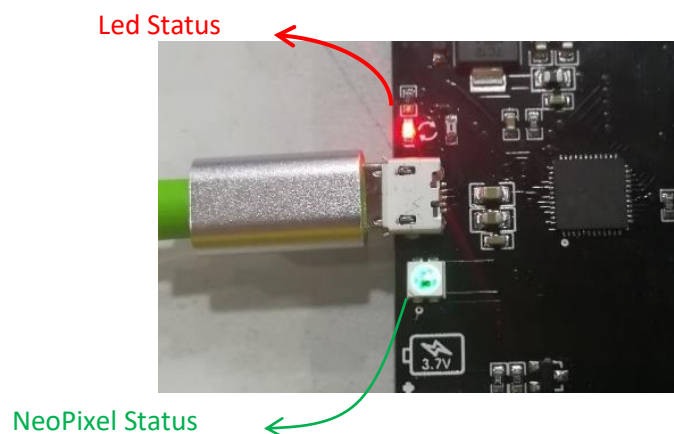


#### ขั้นตอนที่2: กดรีเซ็ต 2 ครั้ง

ในขั้นตอนนี้หลังจากที่เราเชื่อมต่อสาย Micro USB แล้วเราจะเข้าสู่โหมดที่เรียกว่า Mode Bootloader ของตัวบอร์ด KB ROBOT โดยการกดที่ปุ่ม Reset 2 ครั้งติดกัน โดยตัวโหมดนี้จะมิลักษณะเป็น Drive ขึ้นมาใน PC/Laptop ของเรา

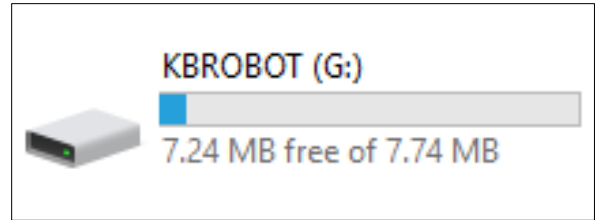
#### ขั้นตอนที่3: การดูสถานะแจ้งเตือน

หลังจากที่เรากดปุ่ม reset เพื่อเข้าสู่โหมด bootloader แล้วก็มีสถานะบอก โดยเราสามารถสังเกตได้จากหลอด NeoPixel ที่ติดอยู่บนตัวบอร์ด KB ROBOT ได้โดยจะแสดงออกมาเป็นแสงสีเขียวและอีก 1 สัญญาณคือหลอด LED ที่อยู่ติดกับหัว Micro USB ก็แสดงออกมาในลักษณะ pulsing ด้วย



#### ขั้นตอนที่4: ตรวจสอบ Drive ใน PC/Laptop

ถ้าการเข้าสู่โหมด bootloader ไม่มีขั้นตอนไหนผิดพลาดเราจะสามารถเห็น Drive ขึ้นมาใหม่ 1 Drive ใน PC/Laptop ของเราโดยมีชื่อว่า KBROBOT เพื่อใช้ในการอัปเดตเฟิร์มแวร์ได้แล้ว



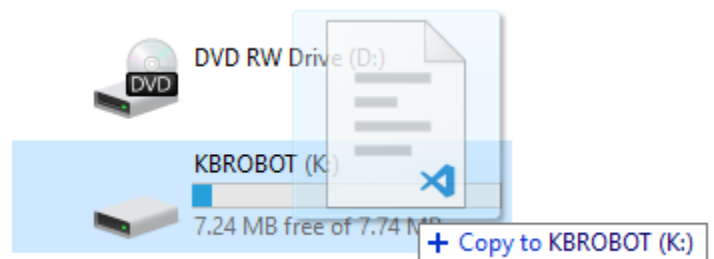
#### ขั้นตอนที่5: Download firmware สำหรับ KB ROBOT

คลิกเพื่อไปยังแหล่ง Download firmware ของ KB ROBOT ได้เลยโดยไฟล์จะเป็นนามสกุล uf2

Click Download

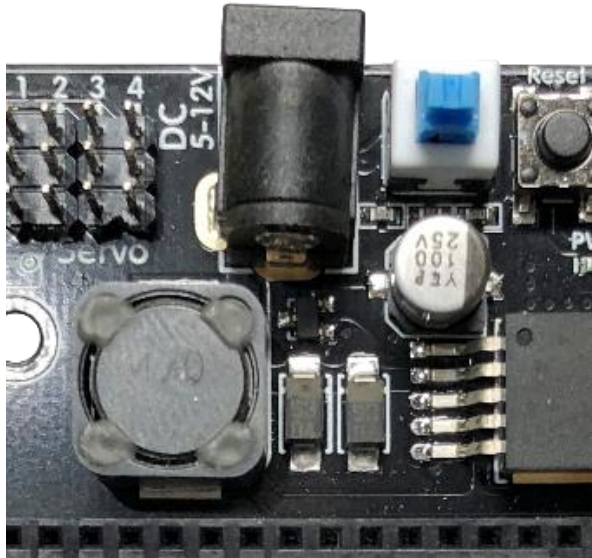
#### ขั้นตอนที่6: ลากไฟล์ UF2 ใส่ใน KB ROBOT

หลังจากที่ download ไฟล์ uf2 มาแล้วให้เราทำการลากไฟล์ uf2 .ใส่ใน drive KBBOOT บน PC/Laptop ของเราได้เลย เพียงเท่านั้น KB ROBOT ของเราก็พร้อมเล่นแล้ว



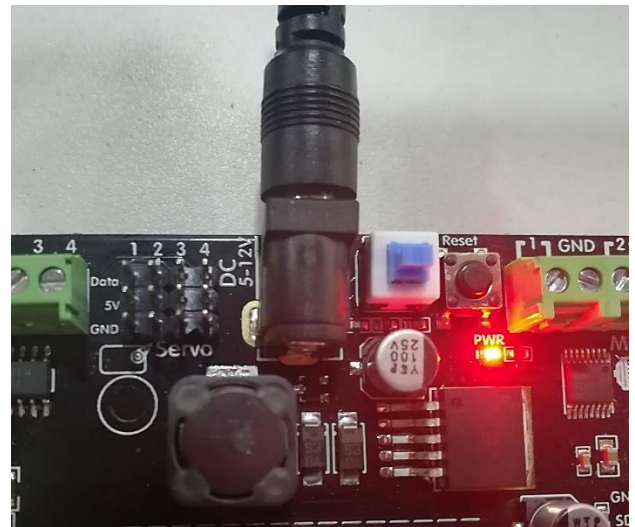
## เริ่มต้นการใช้งาน

### การจ่ายไฟให้ KB ROBOT



ในบอร์ด KB ROBOT ในส่วนของแหล่งจ่ายไฟ  
นั้นจะใช้หัว connector แบบ DC Jack ขนาด 2.1mm (1)  
สามารถรับแรงดันได้ตั้งแต่ 5-12V(2)

เมื่อทำการจ่ายไฟแล้วจะมีไฟแสดงสถานะการทำงานสีแดง  
ขึ้นมา เพื่อแสดงให้เห็นทราบว่าบอร์ดสามารถใช้งานได้ (สามารถกด  
เปิด ปิด การทำงานได้ขากสวิตซ์ข้างๆ)



#### คำแนะนำ

(1): ขนาดของ DC Jack อยู่ที่ 2.1mm

(2): ในกรณีที่ใช้งานทุกฟังก์ชันจะกินพลังงานประมาณ 50W แนะนำให้ใช้ Adapter ดังต่อไปนี้ 5V/2A 9V/2A 12V/1.5A



ตัวอย่าง Adapter

Wall Adapter Switching Power Supply 5.0VDC, 2A, 2.1mm

Item# PSAD-3121

<https://www.gravitechthai.com/product-detail.php?WP=pQqgZKp1GQAgG2rDqYyc4Uuw>

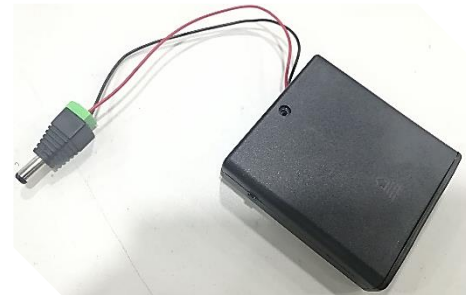
2.1mm Power Jack Breakout Adapter + Battery case AAA x4

Item# COOT-1741

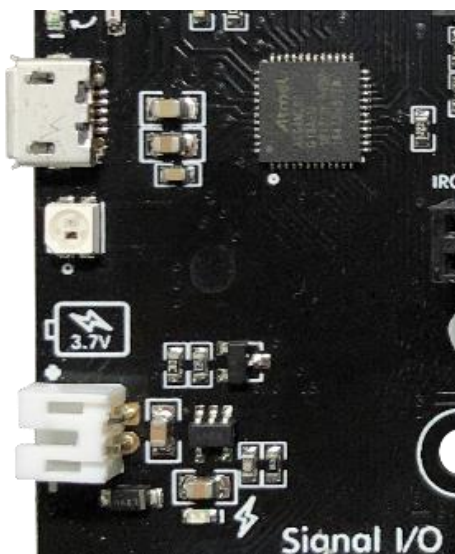
Item# ADBO-0136

<https://www.gravitechthai.com/product-detail.php?WP=pQMgAapIGQWgG2rDqYyc4Uuw>

<https://www.gravitechthai.com/product-detail.php?WP=oGS3BRjkoH9axUF5nrO4Ljo7o3Qo7o3Q>







การจ่ายไฟยังสามารถจ่ายไฟผ่านทางแบตเตอรี่ Lithium Polymer 3.7V(1) ต่อทางหัวต่อ Wafer สีขาวได้ KB ROBOT ก็สามารถใช้งานได้เช่นกัน เนื่องจากเป็นแบตเตอรี่แบบ Li-Po เราจึงไม่ต้องกังวลว่าแบตเตอรี่หมดแล้วหามาเปลี่ยนให้เสียเวลา เพราะเราสามารถทำการรีชาร์จแบตเตอรี่ได้เช่นเดียวกับมือถือ โดยสามารถเสียบสายชาร์จแบบ Micro USB ได้เพื่อชาร์จแบตเตอรี่ได้ และเนื่องจากเป็นแบตเตอรี่ขนาดเล็กเราจึงสามารถนำ KB ROBOT ไปทำโปรเจกในรูปแบบไร้สายได้อย่างสบาย

ในกรณีที่ทำการชาร์จแบตเตอรี่จะมีไฟสถานะสีแดงติดขึ้นมาเพื่อเป็นการบอกว่าการชาร์จแบตเตอรี่อยู่ในระหว่างนี้ไม่ควรใช้งานบอร์ด KB ROBOT เพื่อประสิทธิภาพสูงสุดในการชาร์จแบตเตอรี่ในการชาร์จแต่ละครั้งอาจใช้เวลาในการชาร์จประมาณ 3-4 ชม. ขึ้นอยู่กับปริมาณแบตเตอรี่คงเหลือ ณ ตอนนั้น เมื่อการชาร์จเสร็จสิ้นไฟแสดงสถานะจะดับลงเอง



Lithium Ion Polymer Battery LiPo 1200mAh 3.7V – 703450

Item# PSBA-2976

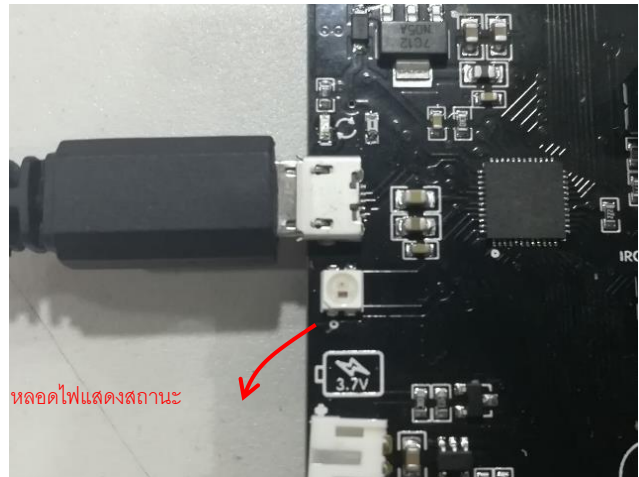
<https://www.gravitechthai.com/product-detail.php?WP=pQEgZ3pmGQAgG2rDqYyc4Uuw>

ข้อแนะนำ

(1): แบตเตอรี่ที่แนะนำให้ใช้ Li-Po 3.7V 1150 - 1200mAh ใช้เวลาชาร์จประมาณ 3-4 hr

## ไฟแสดงสถานะของบอร์ด KB ROBOT

- สีแดงกระพริบคือไฟเลี้ยงอุปกรณ์ไม่พอ
- สีแดงติดค้างคือสาย Micro USB ไม่ได้เชื่อมต่อกับ PC
- สีเขียวคือเข้าสู่โหมด bootloader



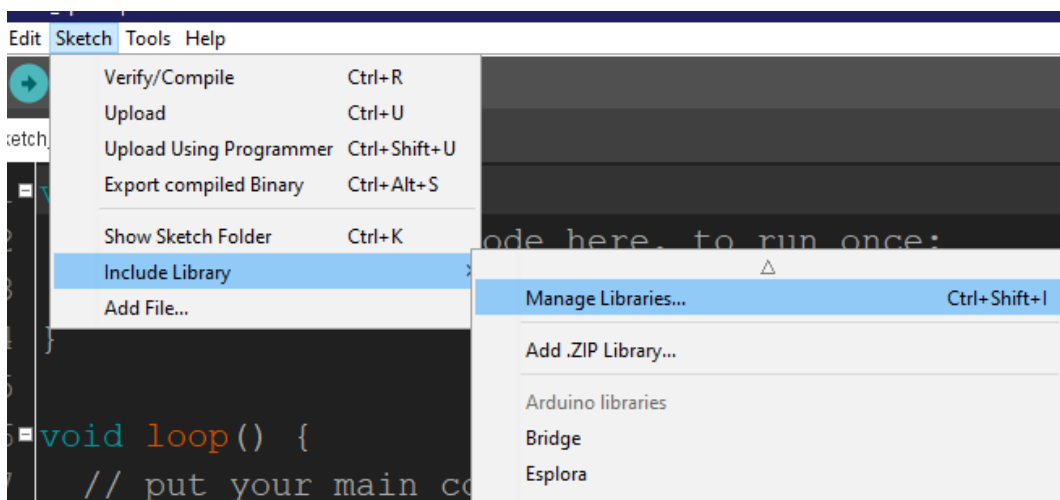
## การใช้งาน Arduino Code กับ KB ROBOT

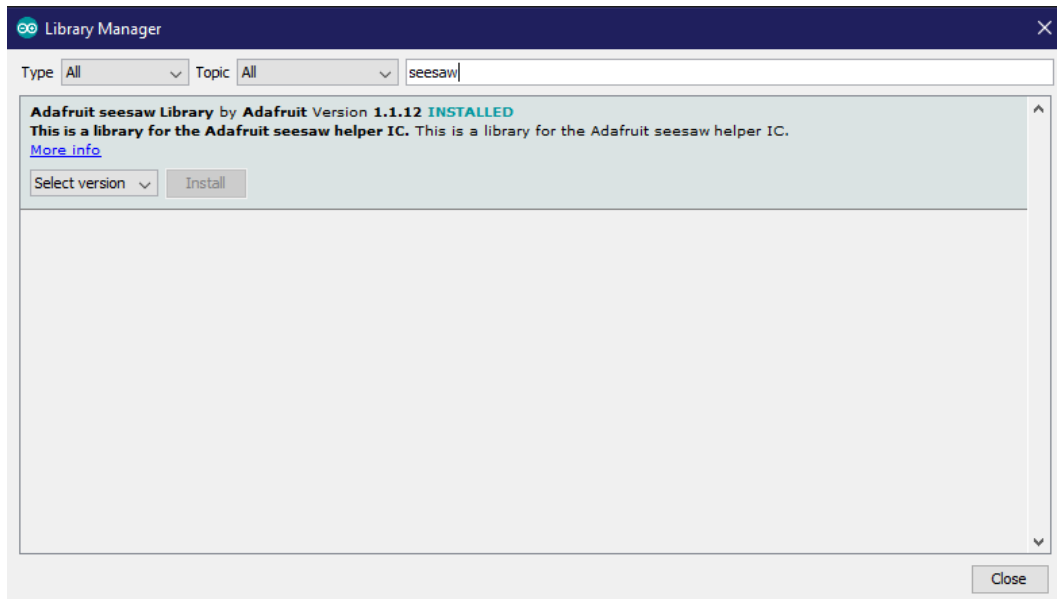
ในการใช้งาน Arduino นั้นจะไม่ได้ทำการควบคุม KB ROBOT โดยตรง แต่จะเป็นการควบคุมผ่านบอร์ดที่ถูกเสียบลงไปบน KB ROBOT ในที่นี้จะใช้เป็น Arduino Nano Chili เพื่อให้สามารถใช้งานได้หลากหลายฟังก์ชัน โดยหลักการทำงานของ KB ROBOT นั้นจะเป็นรูปแบบการสั่งงานจากบอร์ดที่นำมาปลั๊กโดยใช้การสื่อสารในรูปแบบ I2C เป็นหลัก

## ขั้นตอนการติดตั้ง Arduino

สามารถ **Download** แล้วติดตั้งด้วยตัวเองได้จากลิงค์นี้ [Lib Seesaw \(https://github.com/gravitech-engineer/Adafruit\\_Seesaw\)](https://github.com/gravitech-engineer/Adafruit_Seesaw)

หรือ ติดตั้งผ่าน **Library Manager** ได้ตามขั้นตอนต่อไปนี้





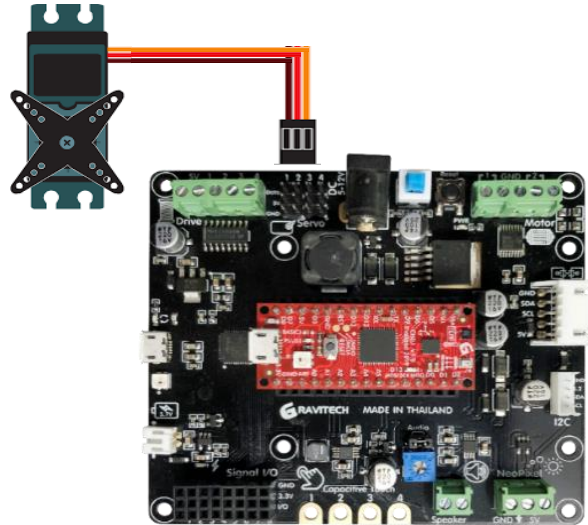
## การใช้งาน Servo Driver Control

ทดสอบการทำงานของ Servo โดยในตัวอย่างจะใช้  
ช่องหมายเลข 1 โดยจะให้ Servo จะขยับ 0 ถึง 180 องศา

เราสามารถควบคุม Servo ได้สูงสุดถึง 4 ตัว โดย  
กำหนดขาดังนี้ ขา 17

( CIRCKIT\_SERVO1 ) , 1 6 ( CIRCKIT\_SERVO2 ) , 1 5

(CIRCKIT\_SERVO3), 14 (CIRCKIT\_SERVO4)



```
#include "Adafruit_Crickit.h"
#include "seesaw_servo.h"

Adafruit_Crickit crickit;
seesaw_Servo myservo1(&crickit);

void setup() {
  Serial.begin(115200);
  if (!crickit.begin()) {
    Serial.println("ERROR!");
    while (1);
  }
  else Serial.println("Crickit started");
  myservo1.attach(CRICKIT_SERVO1);
}

void loop() {
  myservo1.write(0);
  delay(500);
  myservo1.write(180);
  delay(500);
}
```



## คำสั่งที่ใช้ในการควบคุม Servo ที่สามารถใช้ได้

เราสามารถใช้กำหนดค่าสูงสุดต่ำสุดได้โดยใช้คำสั่ง `attach()`

```
myservo.attach(CRICKIT_SERVO1);
```

โดยปกติแล้วค่าดั้งเดิมต่ำสุดถูกกำหนดไว้ที่ 1000 microsec แต่เราจะกำหนดเป็น 750 และค่าสูงสุดจะถูกกำหนดไว้ที่ 2000 microsec แต่เราจะกำหนดใหม่ให้เป็น 2250 แทนตามคำสั่งดังต่อไปนี้

```
myservo.attach(CRICKIT_SERVO1, 750, 2250);
```

สามารถกำหนดค่า pulse เองได้จากคำสั่ง `writeMicroseconds()`

เราสามารถทำการสั่งหยุดการทำงานของ Servo จากคำสั่ง

```
myservo.writeMicroseconds(0);
```

หรือสามารถกำหนดตำแหน่งการหยุดให้แก่ Servo ได้เลยจากคำสั่งเดียวกันโดยจะมีรูปแบบการเขียนดังนี้

```
myservo.writeMicroseconds(number);
```

**number** คือจำนวนความยาวของ pulse ที่กำหนดให้แก่ Servo โดยจะอยู่ในหน่วยของ microsec มีช่วงตั้งแต่ 0(off) ถึง 20000(หมุนไปสุด)



Micro Servo 9g Tower Pro SG90

Item# ROMT-0497

<https://www.gravitechthai.com/product-detail.php?WP=oGA3A0j1oH9axUF5nrO4Ljo7o3Qo7o3Q>

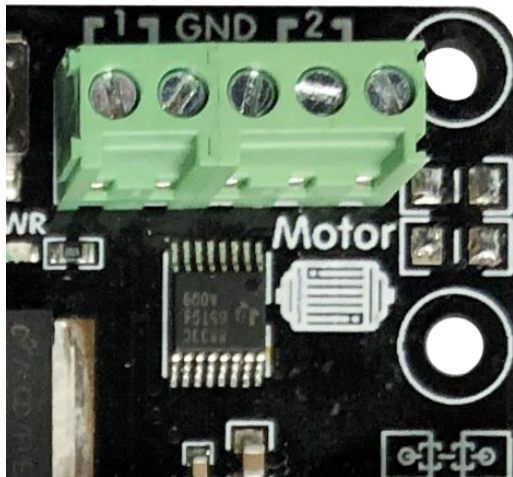


Servo motor 90S

Item# ROMT-1704

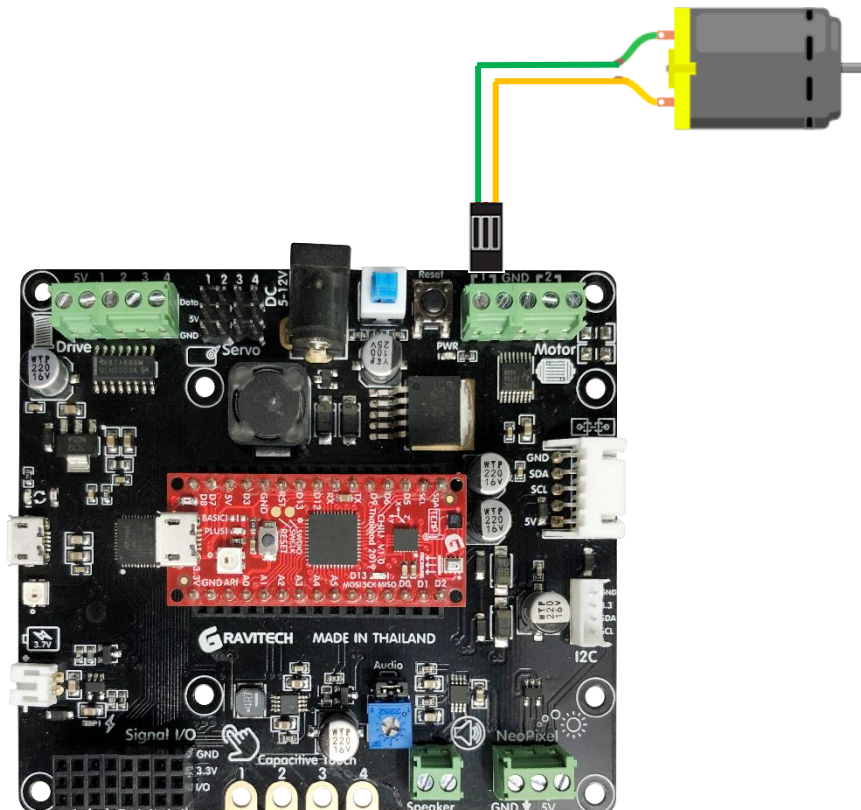
<https://www.gravitechthai.com/product-detail.php?WP=pQlgZap5GQSG2rDqYyc4Uuw>

## การใช้งาน Motor Drive 2CH



**Motor** เป็นอุปกรณ์ที่เป็นพื้นฐานที่สามารถนำไปประยุกต์ใช้ได้มากมาย โดยปกติ **motor** นั้นจะทำงานโดยการจ่ายไฟเข้าไปแล้วจะเกิดการหมุนเกิดขึ้น แต่ปัญหาอยู่ที่การหมุนนั้นมันจะไปทิศทางเดียวเท่านั้นการจะสลับทิศทางการทำงานจะต้องทำการสลับขั้วไฟซึ่งจะเกิดความยุ่งยากเป็นอย่างมาก ดังนั้นในการใช้งาน **motor** เราจึงจะได้เป็นชุดควบคุมการทำงานของ **motor** เป็นส่วนใหญ่เนื่องจากสามารถควบคุมทิศทางการทำงานได้แล้วยังสามารถควบคุมความเร็วในการหมุนได้อีกด้วยจะเป็นเหตุผลที่ว่าทำไมจึงนิยมนำชุดควบคุม **motor** มาใช้งานกัน

ในบอร์ด KB ROBOT นั้นมีวงจรควบคุม **motor** ที่สามารถควบคุม **motor** ได้พร้อมกันถึง 2 ตัวโดยสามารถใช้ในการควบคุมทิศทางการทำงานได้และสามารถควบคุมความเร็วในการหมุนได้เช่นเดียวกัน



```

#include "Adafruit_Crickit.h"
#include "seesaw_motor.h"

Adafruit_Crickit crickit;

seesaw_Motor motor_a(&crickit);
seesaw_Motor motor_b(&crickit);

void setup() {
  Serial.begin(115200);
  Serial.println("Dual motor demo!");

  if(!crickit.begin()){
    Serial.println("ERROR!");
    while(1);
  }
  else Serial.println("Crickit started");

  //attach motor a
  motor_a.attach(CRICKIT_MOTOR_A1, CRICKIT_MOTOR_A2);

  //attach motor b
  motor_b.attach(CRICKIT_MOTOR_B1, CRICKIT_MOTOR_B2);
}

void loop() {
  motor_a.throttle(1);
  motor_b.throttle(-1);
  delay(1000);

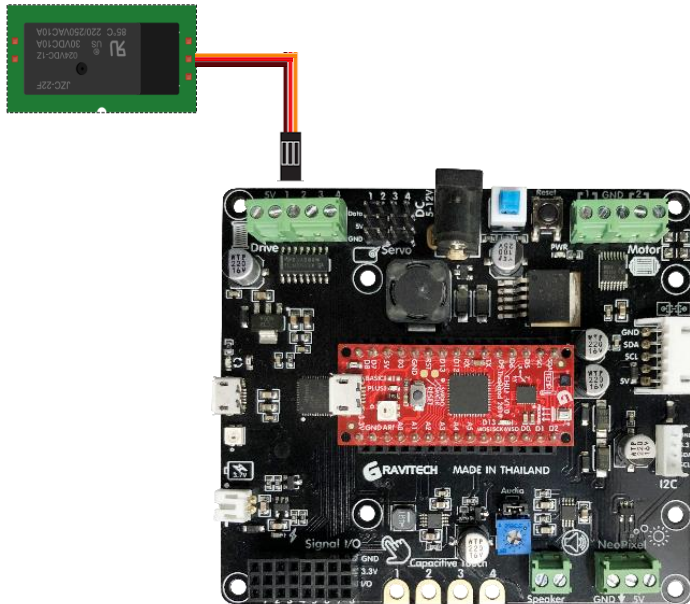
  motor_a.throttle(.5);
  motor_b.throttle(-.5);
  delay(1000);

  motor_a.throttle(0);
  motor_b.throttle(0);
  delay(1000);

  motor_a.throttle(0);
  motor_b.throttle(0);
  delay(500);
}

```

## การใช้งาน High Power Drive



พอดนี้มีความสามารถในการขับอุปกรณ์ที่ใช้กระแสสูงในการทำงานได้

## การต่อใช้งาน High Power Drive

โดยที่พอดนั้นจะมีช่องสำหรับจ่ายไฟ 5V มาแล้ว โดยการเชื่อมต่อนั้นจะต้องต่อไฟและขา Drive1-4 เพื่อควบคุมการทำงานของอุปกรณ์ ขาของ Drive1-4 นั้นมีคุณสมบัติเป็นแบบ PWM เราจึงสามารถควบคุมค่าได้ตั้งแต่ 0 (CRICKIT\_DUTY\_CYCLE\_OFF) ถึง 65535 (CRICKIT\_DUTY\_CYCLE\_MAX)

ในกรณีที่ต้องการใช้ Drive หมายเลขอื่นที่นอกเหนือจากหมายเลข 1 นี้สามารถใช้งานได้ดังนี้ ขา 13 (CRICKIT\_DRIVE1), 12 (CRICKIT\_DRIVE2), 43 (CRICKIT\_DRIVE3), 42 (CRICKIT\_DRIVE4)



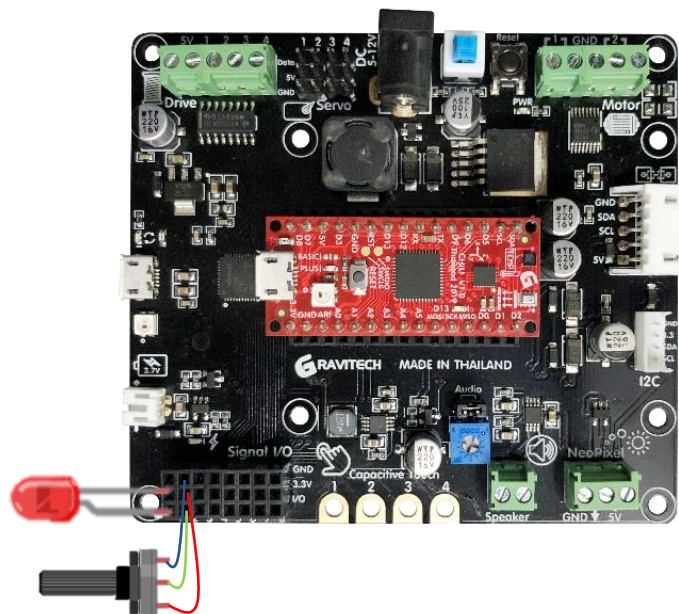
```
#include "Adafruit_Crickit.h"

Adafruit_Crickit crickit;

void setup() {
  Serial.begin(115200);
  Serial.println("1 Drive demo!");
  if(!crickit.begin()){
    Serial.println("ERROR!");
    while(1);
  }
  else Serial.println("Crickit started");
  //our default frequency is 1khz
  crickit.setPWMFreq(CRICKIT_DRIVE1, 1000);
}

void loop() {
  //turn all the way on
  crickit.analogWrite(CRICKIT_DRIVE1, CRICKIT_DUTY_CYCLE_OFF);
  delay(500);
  //turn all the way off
  crickit.analogWrite(CRICKIT_DRIVE1, CRICKIT_DUTY_CYCLE_MAX);
  delay(500);
}
```

## การใช้งาน Signal I/O 8CH



Signal I/O บน KB ROBOT นั้นมีความสามารถเป็น INPUT และ OUTPUT ได้ โดย Logic นั้นจะเป็นแบบ 3.3V โดยสามารถอ่าน INPUT ได้ทั้งแบบ digital และ analog ส่วนการควบคุมนั้นสามารถควบคุมแบบ digital ได้

โดย Signal I/O นั้น KB ROBOT จะใส่ Pin Header แบบตัวเมียมาให้ถึง 8 Channel โดยการใช้งานนั้นสามารถใช้สายไฟแบบ Jumper ตัวผู้ต้องลงไปได้เลย และอีกทั้งยังมีช่องสำหรับจ่ายไฟแบบ 3.3V และ GND ไว้อยู่แล้วสามารถนำไปใช้งานได้ทันที

## หมายเลขของขา Signal I/O

หมายเลขบนบอร์ด	ชื่อที่กำหนดไว้ใน Arduino	หมายเลขขา
1	CRICKIT_SIGNAL1	2
2	CRICKIT_SIGNAL2	3
3	CRICKIT_SIGNAL3	40
4	CRICKIT_SIGNAL4	41
5	CRICKIT_SIGNAL5	11
6	CRICKIT_SIGNAL6	10
7	CRICKIT_SIGNAL7	9
8	CRICKIT_SIGNAL8	8

```

#include "Adafruit_Crickit.h"

Adafruit_Crickit crickit;

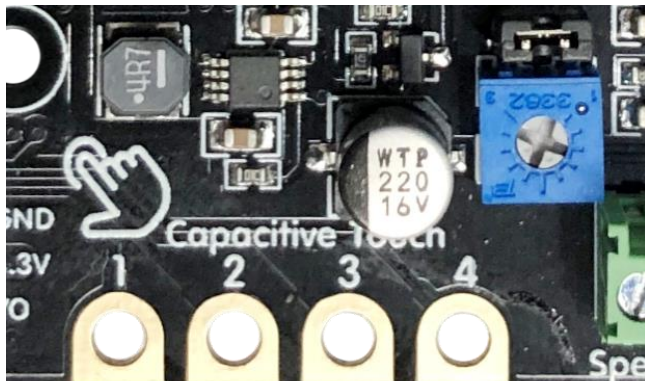
#define BUTTON_1 CRICKIT_SIGNAL1
#define BUTTON_2 CRICKIT_SIGNAL2
#define LED_1 CRICKIT_SIGNAL3
#define LED_2 CRICKIT_SIGNAL4

void setup() {
    Serial.begin(9600);
    if(!crickit.begin()){
        Serial.println("ERROR!");
        while(1);
    }
    else Serial.println("Crickit started");
    //Two buttons are pullups, connect to ground to activate
    crickit.pinMode(BUTTON_1, INPUT_PULLUP);
    crickit.pinMode(BUTTON_2, INPUT_PULLUP);
    // Two LEDs are outputs, on by default
    crickit.pinMode(LED_1, OUTPUT);
    crickit.pinMode(LED_2, OUTPUT);
    crickit.digitalWrite(LED_1, HIGH);
    crickit.digitalWrite(LED_2, HIGH);
}

void loop() {
    if(!crickit.digitalRead(BUTTON_1))
        crickit.digitalWrite(LED_1, HIGH);
    else
        crickit.digitalWrite(LED_1, LOW);
    if(!crickit.digitalRead(BUTTON_2))
        crickit.digitalWrite(LED_2, HIGH);
    else
        crickit.digitalWrite(LED_2, LOW);
}

```

## การใช้งาน Capacitive Touch



Capacitive Touch บน KB ROBOT นั้นเมื่อมีการสัมผัสจากมือของมนุษย์นั้นจะทำการอ่านค่าสัญญาณออกมาในรูปแบบของตัวเลขแบบ analog โดยค่านั้นจะเปลี่ยนแปลงไปตามแรงกดหรือตามวัตถุที่ใช้สัมผัส

```
#include "Adafruit_Crickit.h"

Adafruit_Crickit crickit;

#define CRICKIT_NUM_TOUCH 4
#define CAPTOUCH_THRESH 500

void setup() {
  Serial.begin(9600);
  Serial.println("Cap Touch Demo");
  if(!crickit.begin()) {
    Serial.println("ERROR Starting crickit");
    while(1) ;
  }
  else Serial.println("seesaw started");
}

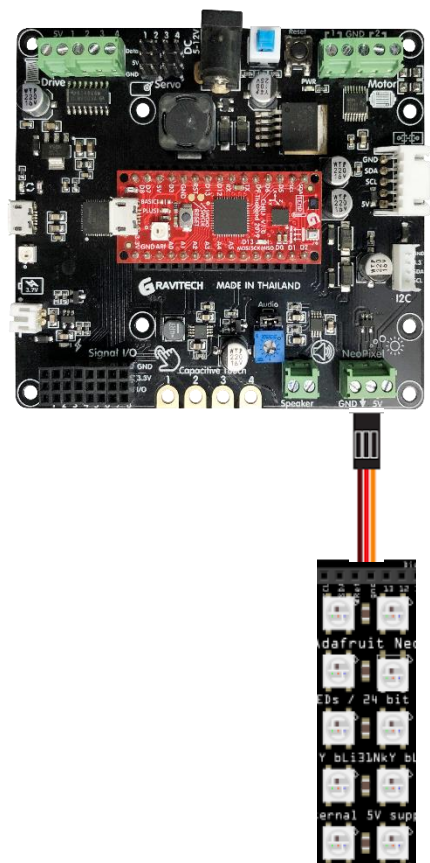
void loop() {

  for(int i=0; i<CRICKIT_NUM_TOUCH; i++){
    uint16_t val = crickit.touchRead(i);

    if(val > CAPTOUCH_THRESH){
      Serial.print("CT");
      Serial.print(i + 1);
      Serial.print(" touched! value: ");
      Serial.println(val);
    }
  }
  delay(100);
}
```



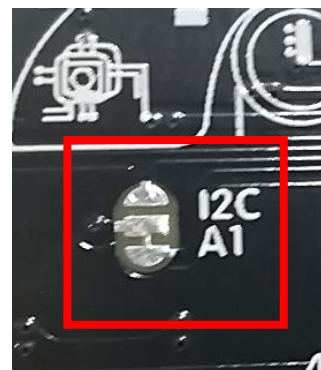
## การใช้งาน NeoPixel



ในการใช้งาน **NeoPixel** นั้นในบอร์ด **KB ROBOT** นั้นจะไม่เหมือนกับการใช้งานฟังก์ชันอื่นที่ผ่านมาเนื่องจากเป็นฟังก์ชันที่ออกแบบมาให้ใช้งานได้ 2 โหมด โดยจะเป็นโหมดที่สามารถควบคุมผ่าน **I2C** ส่วนอีกโหมดจะเป็นการควบคุมผ่าน **Pin** ของบอร์ด **Arduino Nano Chili** โดยตรงดังนั้นหากต้องการนำบอร์ดอื่นที่ไม่ใช่ **Arduino Nano Chili** มาใช้งานจึงจำเป็นต้องตัดลายวงจรและทำการบัดกรีดังลายใหม่ สามารถย้อนดูการตัดลายได้ที่หัวข้อ [Neo Pixel Drive](#)

ในโหมดแรกจะเป็นการต่อแบบใช้ **Arduino Nano Chili** โดยโหมดนี้จะเป็นการควบคุมผ่านขาของ **Arduino Nano Chili** (A1,Pin15) โดยตรงโดยจะใช้ **library Adafruit\_NeoPixel**

ในกรณีที่ให้ **NeoPixel Drive** ผ่าน **I2C** นั้นจำเป็นต้องใช้ **Library** อีกตัวในการควบคุมการทำงานของ **NeoPixel** และขาของ **KB ROBOT** ที่ใช้ในการควบคุม **NeoPixel** นั้นคือขาหมายเลข 20



```
#include <seesaw_neopixel.h>
#define PIN 20

seesaw_NeoPixel strip = seesaw_NeoPixel(10, PIN, NEO_GRB + NEO_KHZ800);

void setup() {
  Serial.begin(115200);
  Wire.begin(4,5);
  if (!strip.begin()) {
    Serial.println("ERROR");
    while (1);
  }
  Serial.println("seesaw started!");
  strip.show(); // Initialize all pixels to 'off'
}

void loop() {
  for(int i=0;i<NUMPIXELS;i++){
    strip.setPixelColor(i, strip.Color(0,150,0));
    strip.show();
    delay(50);
  }
}
```



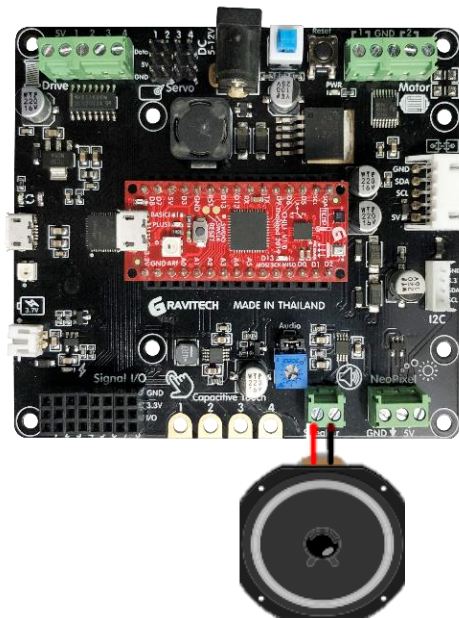
สามารถใช้ **NeoPixel** แบบในรูปได้

Gerora

Item# MISP-0382

<https://www.gravitechthai.com/product-detail.php?WP=oGI3ZRj0oH9axUF5nrO4Ljo7o3Qo7o3Q>

## การใช้งาน Speaker Drive



อย่างที่กล่าวไว้ข้างต้นว่า KB ROBOT นั้นนอกจากจะสามารถตั้งสีสันจากหลอดไฟได้แล้วยังสามารถควบคุมเสียงได้ด้วย ตัวอย่างนี้จะเป็นการควบคุมผ่านขา (A0, Pin14)

ตัวอย่างส่งเสียงบีป

```
int speaker = 14;
void setup(void)
{
  pinMode(speaker, OUTPUT); //buzzer
}
void loop()
{
  unsigned char i;
  for (i = 0; i < 100; i++)
  {
    digitalWrite(speaker, HIGH);
    delay(1); //wait for 1ms
    digitalWrite(speaker, LOW);
    delay(1); //wait for 1ms
  }
}
```

หรือสามารถ Download Code ตัวอย่างการเล่นเป็นเพลงได้จาก [Link นี้](#)

ตัวอย่างลำโพงที่สามารถใช้งานได้



8x53mm 8 Ohm Speaker with 2.00mm Connector

Item# COOT-2716

<https://www.gravitechthai.com/product-detail.php?WP=pQMgZ3pkGQAgG2rDqYyc4Uuw>

17mm 8 Ohm Speaker 7.56mm Pitch - Breadboard

Item# COOT-2709

<https://www.gravitechthai.com/product-detail.php?WP=pQugZ3pkGQAgG2rDqYyc4Uuw>





## Download

สามารถ Download Library ได้จากทางลิงค์ข้างใต้

Library: [https://github.com/gravitech-engineer/Adafruit\\_Seesaw](https://github.com/gravitech-engineer/Adafruit_Seesaw)

Example: [https://github.com/gravitech-engineer/KB\\_ROBOT](https://github.com/gravitech-engineer/KB_ROBOT)