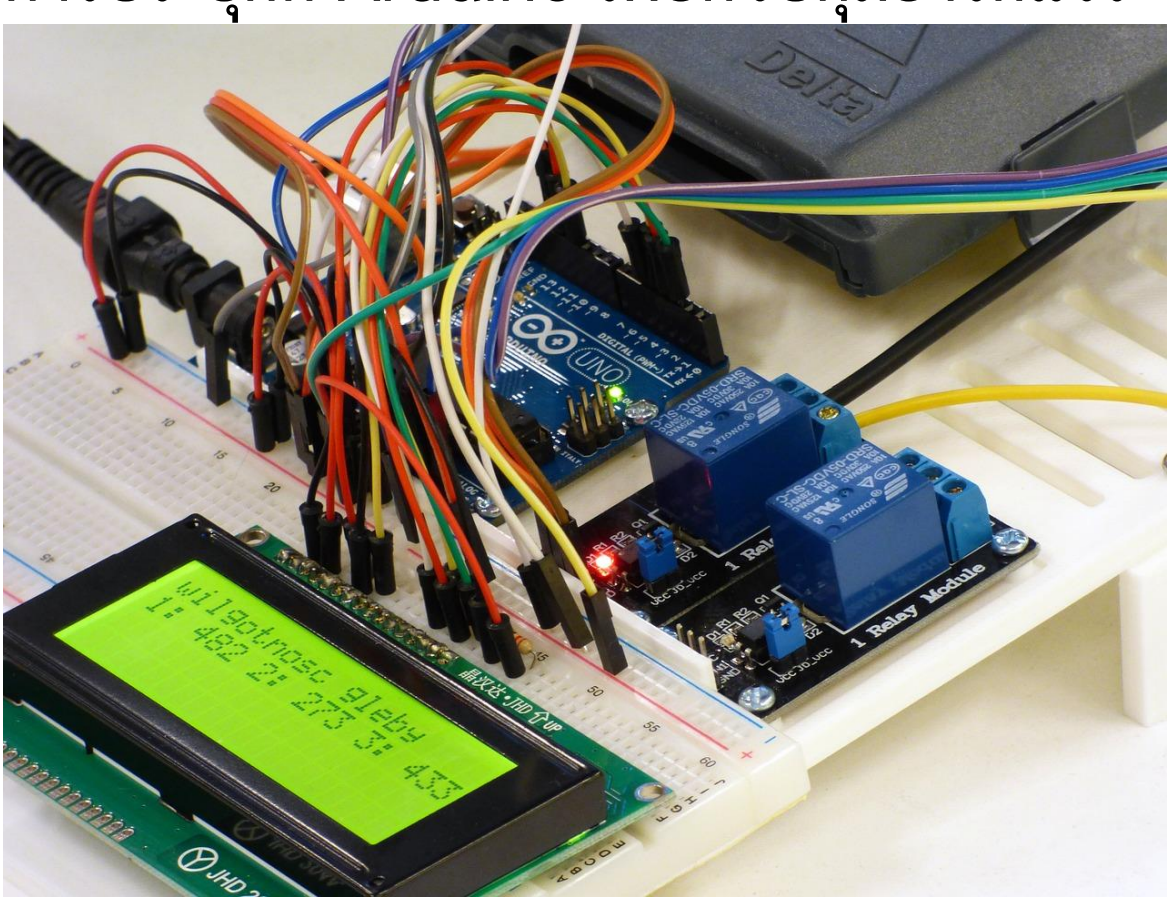
การประยุกต์ Arduino เพื่อควบคุมฮาร์ดแวร์



ผศ.ดร.วิโรจน์ จงชนะชววัฒน์

สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม
มหาวิทยาลัยราชภัฏเพชรบุรี

การประยุกต์ Arduino เพื่อควบคุมฮาร์ดแวร์

ผศ.ดร.วิโรจน์ จงชนะชววัฒน์

สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม

มหาวิทยาลัยราชภัฏเพชรบุรี

การประยุกต์ Arduino เพื่อควบคุมฮาร์ดแวร์

วิโรจน์ จงชนะชววัฒน์

พิมพ์ครั้งที่ 31 กรกฎาคม 2567 จำนวน 100 เล่ม

ข้อมูลทางบรรณานุกรมหอสมุดแห่งชาติ

วิโรจน์ จงชนะชววัฒน์.

การประยุกต์ Arduino เพื่อควบคุมฮาร์ดแวร์.—กรุงเทพฯ:ทวีผลอินเตอร์พริ้นท์, 2567, 149หน้า

1.การประยุกต์ Arduino เพื่อควบคุมฮาร์ดแวร์.

I.ชื่อเรื่อง

ISBN:

พิมพ์โดย

บริษัท ทวีผลอินเตอร์พริ้นท์ จำกัด

ที่อยู่ 416 ซ.สุขุมวิท50 แขวงพระโขนง เขตคลองเตย กรุงเทพฯ 10260

คำนำ

หนังสือการประยุกต์ Arduino เพื่อควบคุมฮาร์ดแวร์เล่มนี้ เป็นการเรียบเรียงเนื้อหาให้กระชับในการศึกษา จึงเริ่มที่พัฒนาการบอร์ด Arduino โปรแกรมที่ใช้ในการเขียนให้บอร์ด Arduino ทำงาน การควบคุมขาอินพุทและเอาท์พุท แล้วจึงนำไปควบคุมเซ็นเซอร์ จนกระทั่งนำมาประยุกต์กับโพรเจกต์ต่างๆ จึงยิ่งทำให้เพิ่มความรู้ และความเชี่ยวชาญในการควบคุมฮาร์ดแวร์ ทั้งภาคทฤษฎีและภาคปฏิบัติได้อย่างแจ่มแจ้ง

หนังสือเล่มนี้ ผู้เขียนได้พยายามเขียนมีเนื้อหา เพื่อให้กระชับ ซึ่งจะทำให้ผู้อ่านได้เข้าสู่เนื้อหาอย่างรวดเร็ว แต่อย่างไรก็ตาม มีคำกล่าวเสมอว่า ทฤษฎี และการปฏิบัติย่อมมีผลการทำงานที่ไม่เหมือนกันทีเดียว ในการเขียนโปรแกรมควบคุมฮาร์ดแวร์ ก็จำเป็นอย่างยิ่งที่ต้องศึกษาในเชิงปฏิบัติ เพราะว่าจะทำให้นักเขียนโปรแกรมมีความเข้าใจ คำสั่ง ตลอดจนถึงการออกแบบโปรแกรมให้สามารถควบคุมฮาร์ดแวร์ได้ตรงความต้องการ

ผู้เขียนต้องขอกราบขอบพระคุณต่อนายเจี๊ยบ แซ่จึงและนางเคี้ยม แซ่ลี้ ซึ่งเป็นบิดามารดาของผู้เขียน แม้ว่าท่านทั้งสอง ไม่ได้รับการศึกษาที่สูงก็ตาม แต่ท่านทั้งสองคือต้นแบบที่เปรียบเสมือนคุณครูคนแรกที่ได้สอนสั่งให้ผู้เขียนเรียนรู้ ตลอดจนเป็นแรงบันดาลใจในการพัฒนาตนเอง ตั้งแต่วัยทารก เติบใหญ่ขึ้นจนถึงปัจจุบัน รวมทั้งครูบาอาจารย์ที่ได้แนะนำสอนสั่ง จนผู้เขียนมีความรู้ความเข้าใจ ในการเรียนรู้ทำงาน ชีวิตและสังสารวัฏดังนี้ พระอาจารย์สิริ คุณากโร, รศ.ดร.กอบชัย เดชหาญ, รศ.ดร.มนตรี คำเงิน, รศ.ดร.อิทธิพงศ์ ชัยสายัณห์, รศ. ช. ณิฐ์ศิริ สุยสุวรรณ, ศ. สุพัฒน์ บุณยฤทธิกิจ และ ศ.ดร.ปิติเขต สู้รักษา ตลอดจนคณาจารย์และผู้มีพระคุณทุกท่าน ที่สอนสั่งตั้งแต่วัยเด็ก จนถึงปัจจุบัน ที่มิได้เอ่ยนามมา ณ ที่นี้ด้วย

สุดท้ายนี้ หนังสือเล่มนี้ ถือเป็นของขวัญในวันเกิดของปีนี้ ซึ่งเป็นประวัติศาสตร์ที่น่าจดจำของผู้เขียนอีกคำรบหนึ่ง ผู้เขียนหวังว่า หนังสือเล่มนี้ จะเป็นประโยชน์ต่อผู้ศึกษาหรือผู้สนใจทั่วไป หากมีข้อเสนอแนะที่ช่วยให้หนังสือฉบับนี้มีความสมบูรณ์ยิ่งขึ้น ผู้เขียนยินดีน้อมรับด้วยความขอบคุณ ทั้งนี้เพื่อให้เนื้อหาความรู้นี้ เสริมสร้างทักษะการเรียนรู้ด้านการเขียนโปรแกรม รวมทั้งการประยุกต์ Arduino เพื่อควบคุมฮาร์ดแวร์ ได้อย่างมีประสิทธิภาพยิ่ง ๆ ขึ้นไป

วิโรจน์ จงชนะชววัฒน์
“บ้านแห่งสังสารวัฏ”
กรกฎาคม 2567

สารบัญ

สารบัญ(ต่อ)

หน้า

สารบัญ(ต่อ)

หน้า

สารบัญ(ต่อ)

หน้า

สารบัญ(ต่อ)

สารบัญรูป

รูปที่ **หน้า**

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

สารบัญรูป(ต่อ)

รูปที่ **หน้า**

สารบัญรูป(ต่อ)

รูปที่ **หน้า**

สารบัญรูป(ต่อ)

รูปที่ **หน้า**

สารบัญตาราง

บทที่ 1 พัฒนาการและเทคโนโลยีบอร์ด Arduino

บอร์ด Arduino มีกำเนิดที่ สถาบันการออกแบบอินเทอร์แอคชันอีเวรีย(Ivrea Interaction Design Institute) ในอิตาลี ในปี 2005 โดยมัสซิโม บันซี่ (Massimo Banzi) และดาวิด กวาร์เทียลเลส (David Cuartielles) เป็นผลงานของทีมผู้สร้างและนักวิจัยที่มีความสนใจในการสร้างแพลตฟอร์มอิเล็กทรอนิกส์ที่ง่ายต่อการใช้งานและการเรียนรู้สำหรับนักพัฒนาและนักวิจัยในด้านอิเล็กทรอนิกส์และการโปรแกรม ลดความยากลำบากในการใช้งานและเข้าใจบอร์ดการควบคุมอิเล็กทรอนิกส์ที่มีอยู่ในช่วงนั้น

ในยุคนั้น ต้องเผชิญกับความซับซ้อนของวงจรและโปรแกรม ทำให้นักพัฒนาผู้ที่เพิ่งเริ่มมีประสบการณ์มีความยากลำบากในการทดลองและสร้างโพรเจกต์ Arduino มีการออกแบบให้เป็นเครื่องมือที่ใช้งานได้ง่ายสำหรับทุกคน โดยการนำเสนอบอร์ดที่มีขาอินพุตและขาเอาต์พุตที่มีหมายเลขประจำอินพุต/เอาต์พุตที่ช่วยให้นักพัฒนาสามารถเชื่อมต่ออุปกรณ์และเซ็นเซอร์อิเล็กทรอนิกส์ได้อย่างง่าย Arduino IDE พัฒนาขึ้นเพื่อเป็นแอปพลิเคชันโปรแกรมที่ใช้งานสะดวก สำหรับการเขียนและแปลโปรแกรม และอัปโหลดไปยังบอร์ด

ด้วยเหตุนี้ การได้รับความนิยมในการสร้างโพรเจกต์อิเล็กทรอนิกส์และการควบคุมฮาร์ดแวร์ไปทั่วโลก รวมทั้งยังเป็นโพรเจกต์โอเพนซอร์ส(Open source) ที่เปิดให้นักพัฒนาและนักวิจัยมีโอกาสในการปรับปรุงและพัฒนาให้ดียิ่งขึ้น สร้างปรากฏการณ์ในการพัฒนาและการสร้างสรรค์ในด้านอิเล็กทรอนิกส์และการเขียนโปรแกรมในปัจจุบัน เป็นแพลตฟอร์มฮาร์ดแวร์และซอฟต์แวร์ที่ออกแบบมาเพื่อใช้ในการพัฒนาโพรเจกต์อิเล็กทรอนิกส์และการเขียนโปรแกรม โดยจะสามารถอธิบายพัฒนาการและเทคโนโลยีได้ ดังนี้

1.1 พัฒนาการบอร์ด Arduino

บอร์ด Arduino มีหลายรุ่นและรูปแบบต่าง ๆ ที่สามารถเลือกใช้ตามความต้องการ ผู้เขียนขอนำมาเขียนแยกในแต่ละตระกูลหลักๆ ที่ทำให้ผู้อ่านเกิดความเข้าใจภาพรวมของบอร์ด Arduino รุ่นหลักๆ ได้ ซึ่งมีรายละเอียดได้ ดังนี้

1.1.1 บอร์ดตระกูล Arduino Uno

เป็นหนึ่งตระกูลดั้งเดิม โดยในรุ่นนี้ ที่ได้รับความนิยมมากที่สุดและใช้งานอย่างกว้างขวางในโพร-เจกต์อิเล็กทรอนิกส์และการโปรแกรม ซึ่งในตระกูลนี้ จะประกอบไปด้วยบอร์ดดังต่อไปนี้

1.Leonardo

2.Micro

3.UNO Mini Limited Edition

4.UNO R4 Minima

5.UNO R4 WiFi

6.UNO R3

7.UNO R3 SMD

8.UNO WiFi Rev2

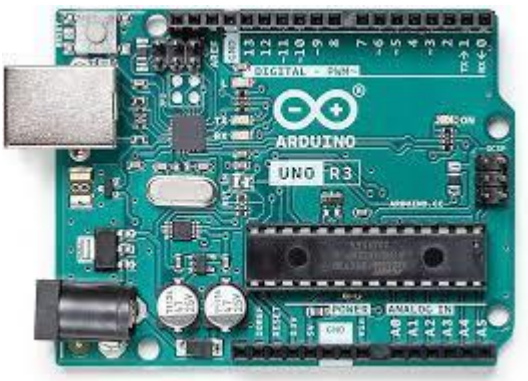
9.Yún Rev2

10.Zero

ซึ่งแต่ละรุ่น จะมีคุณสมบัติที่แตกต่างกัน ในที่หนังสือนี้ ผู้เขียนขอนำมากล่าวเฉพาะบางบอร์ดของรุ่นนี้ คือ Arduino UNO R3 และ Arduino UNO R4 WiFi

1.1.1.1 Arduino UNO R3

เป็นบอร์ดที่เหมาะสำหรับการเริ่มใช้งานบอร์ด Arduino ซึ่งแสดงได้ดังรูปที่ 1.1

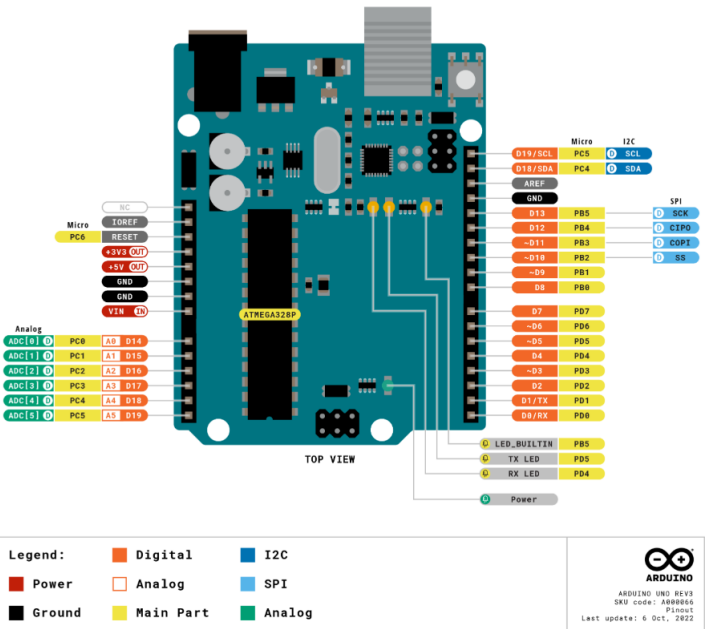


https://store.arduino.cc/products/arduino-uno-rev3: Accessed: May5, 2024.

รูปที่ 1.1 บอร์ด Arduino UNO

Arduino UNO เป็นบอร์ดไมโครคอนโทรลเลอร์ที่มี ATmega328P มี 14 ขาดิจิทัลอินพุต/เอาต์พุต (ซึ่ง 6 ขาสามารถใช้เป็นเอาต์พุต การปรับความกว้างของพัลส์(PWM) ได้), 6 ขาอนาล็อกอินพุต/เอาต์พุต, 16 MHz, การเชื่อมต่อยูเอสบี (USB), หัวต่อเสียงไฟ, หัวต่อ ICSP และ ปุ่มรีเซ็ต (Reset Button) ผู้อ่านเพียงแค่เชื่อมต่อกับคอมพิวเตอร์ด้วยสายยูเอสบี หรือให้พลังงานผ่านอะแดปเตอร์ AC-to-DC หรือแบตเตอรี่ เพื่อเริ่มต้นใช้งาน จากรูปที่ 1.2 สามารถอธิบายคุณสมบัติและข้อมูลสำคัญเกี่ยวกับบอร์ด Arduino Uno ได้ดังนี้

1. ไมโครคอนโทรลเลอร์ (Microcontroller) Arduino Uno ใช้ไมโครคอนโทรลเลอร์โดยประกอบด้วยชิป ATmega328P ซึ่งเป็นไมโครคอนโทรลเลอร์ความเร็ว 16 MHz ที่สามารถควบคุมขาเอาต์พุตและอินพุตได้อย่างสมบูรณ์ และมีความจุของแฟลชเมมโมรี่ 32 KB สำหรับโปรแกรม
2. ขาอินพุต/เอาต์พุต (I/O Pins) Arduino Uno มีขาอินพุตและขาเอาต์พุตรวมทั้งหมด 14 ขา (digital I/O pins) โดยในนั้นมี 6 ขาเอาต์พุตที่สามารถใช้เป็น การปรับความกว้างของพัลส์ได้ นอกจากนี้ยังมีขาสำหรับการสื่อสารแบบ UART (Serial), I2C, และ SPI



https://store.arduino.cc/products/arduino-uno-rev3: Accessed: May5, 2024.

รูปที่ 1.2 ขาของบอร์ด Arduino UNO

3. แหล่งจ่ายไฟ (Power Supply) Arduino Uno สามารถใช้งานได้ในการจ่ายไฟในช่วง 7-12 โวลต์ (V) โดยใช้เส้นสายแบบดีซี(DC) หรือในบางกรณีสามารถใช้ยูเอสบี เป็นแหล่งจ่ายไฟ

4. การเชื่อมต่อยูเอสบี Arduino Uno มาพร้อมพอร์ตยูเอสบี ที่สามารถใช้ในการเขียนโปรแกรมและอัปโหลดไฟล์ต่าง ๆ จากคอมพิวเตอร์ไปยังบอร์ด

5. ความจุแฟลชและ SRAM ไมโครคอนโทรลเลอร์ ATmega328P บน Arduino Uno มีความจุแฟลชเมมโมรี่ 32KB และ SRAM 2KB ที่สามารถใช้ในการเก็บโปรแกรมและข้อมูลในรันไทม์ตามลำดับ

6. สวิตช์ตั้งใหม่ (Switch Reset) บอร์ดมีสวิตช์ที่สามารถใช้ในการรีเซ็ตบอร์ด Arduino เพื่อให้โปรแกรมเริ่มทำงานใหม่

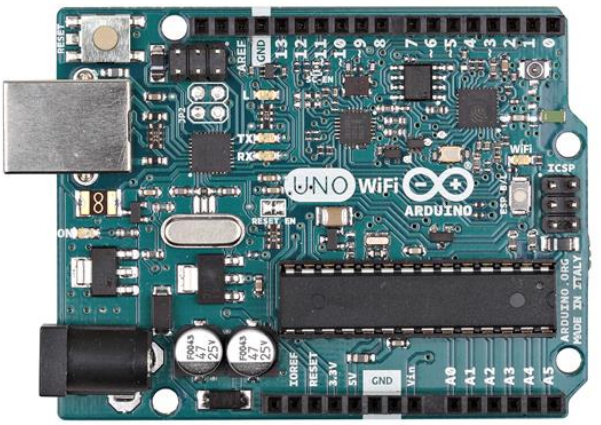
7. บอร์ดสำรองสำหรับแทนที่ Arduino Uno มีบอร์ดสำรองสำหรับแทนที่ (Shield) ที่สามารถใส่ลงบนบอร์ดหลักเพื่อเพิ่มความสามารถและการเชื่อมต่ออุปกรณ์เสริม

8. ส่วนประกอบ Arduino Uno สามารถใช้งานได้บนหลายระบบปฏิบัติการ อาทิ Windows, Mac OS, และ Linux โดยมี Arduino IDE ที่รองรับได้หลายแพลตฟอร์ม

Arduino Uno เป็นรุ่นที่เหมาะสำหรับนักพัฒนาที่เริ่มต้นในการทดลองและสร้างโพรเจกต์อิเล็กทรอนิกส์และการเขียนโปรแกรม ด้วยความสามารถในการเชื่อมต่ออุปกรณ์และเซ็นเซอร์ต่าง ๆ รวมถึงมีชุดความสามารถที่ใช้งานง่าย สำหรับการเรียนรู้

1.1.1.2 Arduino UNO R4 Wi-Fi

บอร์ด Arduino UNO Wi-Fi คือ หนึ่งในบอร์ดที่นำเสนอความสามารถในการเชื่อมต่ออินเทอร์เน็ตไร้สาย (Wi-Fi) พร้อมกับความสะดวกในการใช้งานของบอร์ด Arduino UNO แสดงได้ดังรูปที่ 1.3 ซึ่งเป็นบอร์ดที่ทำงานเช่นเดียวกันกับบอร์ด Arduino UNO มีคุณสมบัติพิเศษในเรื่องการเชื่อมต่ออินเทอร์เน็ต Wi-Fi ซึ่งจะมีชิปเสริม Wi-Fi ภายในตัวบอร์ด

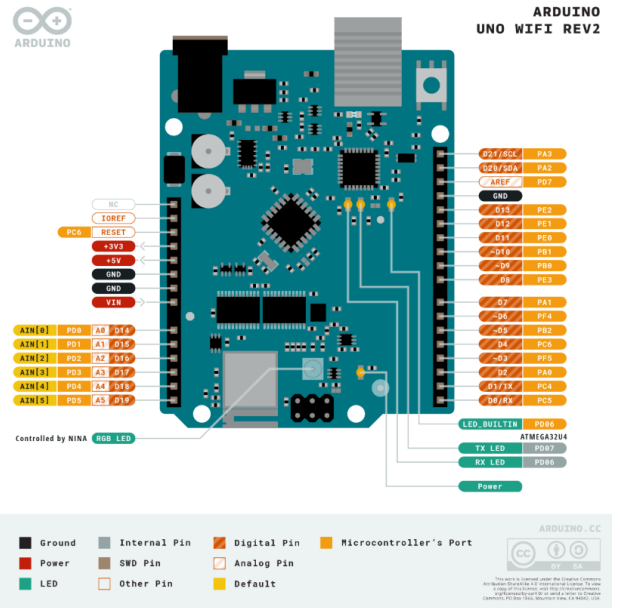


https://store.arduino.cc/products/arduino-uno-wifi-rev2: Accessed: May5, 2024.

รูปที่ 1.3 บอร์ด Arduino UNO Wi-Fi

จากรูปที่ 1.4 จะแสดงคุณสมบัติและข้อมูลสำคัญเกี่ยวกับบอร์ด Arduino Uno Wi-Fi ได้ดังนี้

1. ไมโครคอนโทรลเลอร์ (Microcontroller) Arduino Uno Wi-Fi ใช้ไมโครคอนโทรลเลอร์ ATMega328P อย่างเดียวกับ Arduino Uno ซึ่งเป็นไมโครคอนโทรลเลอร์ความเร็ว 16 MHz ที่มีความจุแฟลชเมมโมรี่ 32 KB และ SRAM 2 KB สำหรับการจัดการโค้ดและข้อมูล
2. ขาอินพุต/เอาต์พุต Arduino Uno Wi-Fi มีขาอินพุตและขาเอาต์พุตรวมทั้งหมด 14 ขา โดยในนั้นมี 6 ขาเอาต์พุตที่สามารถใช้เป็น การปรับความกว้างของพัลส์ได้ นอกจากนี้ยังมีขาสำหรับการสื่อสารแบบ UART (Serial), I2C, และ SPI
3. การเชื่อมต่อยูเอสบี Arduino Uno Wi-Fi มาพร้อมพอร์ตยูเอสบี สำหรับการเขียนโปรแกรมและอัปโหลดไฟล์ต่าง ๆ จากคอมพิวเตอร์ไปยังบอร์ด



https://store.arduino.cc/products/arduino-uno-wifi-rev2: Accessed: May5, 2024.

รูปที่ 1.4 ขาของบอร์ด Arduino UNO Wi-Fi

4. การเชื่อมต่อ Wi-Fi Arduino Uno Wi-Fi มาพร้อมชิป Wi-Fi (ATWINC1500) ภายในตัวบอร์ด เป็นวิธีที่สะดวกในการเชื่อมต่อกับเครือข่าย Wi-Fi ทำให้เราสามารถส่งข้อมูลไร้สายไปยังอินเทอร์เน็ต, ควบคุมอุปกรณ์, หรือเข้าถึงข้อมูลแบบออนไลน์

5. ความสามารถในการจัดเก็บไฟล์ Arduino Uno Wi-Fi มีหน่วยความจำ (EEPROM) ขนาด 1 KB สำหรับการจัดเก็บข้อมูลและการจัดการการบันทึกข้อมูล

6. การจ่ายไฟ Arduino Uno Wi-Fi สามารถใช้งานได้ในการจ่ายไฟในช่วง 7-12 โวลต์ (V) โดยใช้เส้นสายแบบดีซี หรือในบางกรณีสามารถใช้ยูเอสบีเป็นแหล่งจ่ายไฟ

7. สวิตช์เริ่มใหม่ บอร์ดมีสวิตช์ที่สามารถใช้ในการรีเซ็ตบอร์ด Arduino เพื่อให้โปรแกรมเริ่มทำงานใหม่

8. ส่วนประกอบ Arduino Uno Wi-Fi สามารถใช้งานได้บนหลายระบบคอมพิวเตอร์รวมถึง Windows, Mac OS, และ Linux โดยมี Arduino IDE ที่รองรับหลายแพลตฟอร์ม

Arduino Uno Wi-Fi เหมาะสำหรับโพรเจกต์ที่ต้องการการเชื่อมต่อกับอินเทอร์เน็ต Wi-Fi เพื่อควบคุมหรือจัดการข้อมูลออนไลน์ มันเป็นเครื่องมือที่ดีสำหรับโพรเจกต์อินเทอร์เน็ตประสานสรรพสิ่งและโพรเจกต์ที่ต้องการการเชื่อมต่อไร้สายเพื่อสื่อสารและควบคุมอุปกรณ์แบบไร้สายผ่านเครือข่าย Wi-Fi ยังมีการพัฒนาบอร์ด Arduino ขึ้นมาอีกหลายแบบ เพื่อให้มีประสิทธิภาพในการใช้งาน ผ่าน Wi-Fi และรองรับการประยุกต์ใช้งาน ผ่านระบบอินเตอร์เน็ตสรรพสิ่งให้ยิ่ง ๆ ขึ้นไป ซึ่งถ้าผู้อ่านสนใจ ก็สามารถหารายละเอียดรุ่นอื่น ๆ เพิ่มเติมได้

1.1.2 ตระกูลบอร์ด Arduino Nano

Arduino Nano คือหนึ่งในบอร์ด Arduino ซีรีส์อย่างหนึ่งที่มีขนาดเล็กและเหมาะสำหรับโพรเจกต์ที่ต้องการบอร์ดควบคุมที่มีขนาดเล็กและทรงพลังสำหรับโพรเจกต์ที่จำเป็นต้องมีความพกพาง่าย ซึ่งในตระกูลนี้จะประกอบไปบอร์ดดังต่อไปนี้

1.Nano

2.Nano 33 BLE

3.Nano 33 BLE Sense

4.Nano 33 BLE Sense Rev2

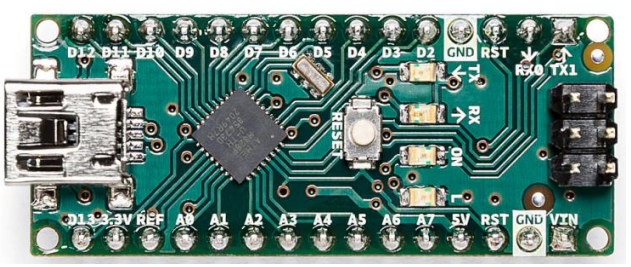
5.Nano 33 IoT

6.Nano ESP32

7.Nano Every

8.Nano RP2040 Connect

ผู้เขียนขอกล่าวเฉพาะรุ่น Arduino Nano เท่านั้น ซึ่ง Arduino Nano เป็นอุปกรณ์ที่เหมาะสำหรับนักพัฒนาและผู้สนใจในการสร้างโพรเจกต์อิเล็กทรอนิกส์หรือโปรแกรม Arduino ที่มีขนาดเล็ก แสดงได้ดังรูปที่ 1.5



https://store.arduino.cc/products/arduino-nano: Accessed: May5, 2024.

รูปที่ 1.5 บอร์ด Arduino Nano

จากรูปที่ 1.6 จะแสดงคุณสมบัติหลักของ Arduino Nano ได้ดังนี้

1.ขนาดเล็ก Arduino Nano มีขนาดเล็กและสามารถประมาณเท่ากับบัตรเครดิตหรือเล็กกว่า มีขนาดเล็กมากเหมาะสำหรับการใช้ในโพรเจกต์ที่มีพื้นที่จำกัด

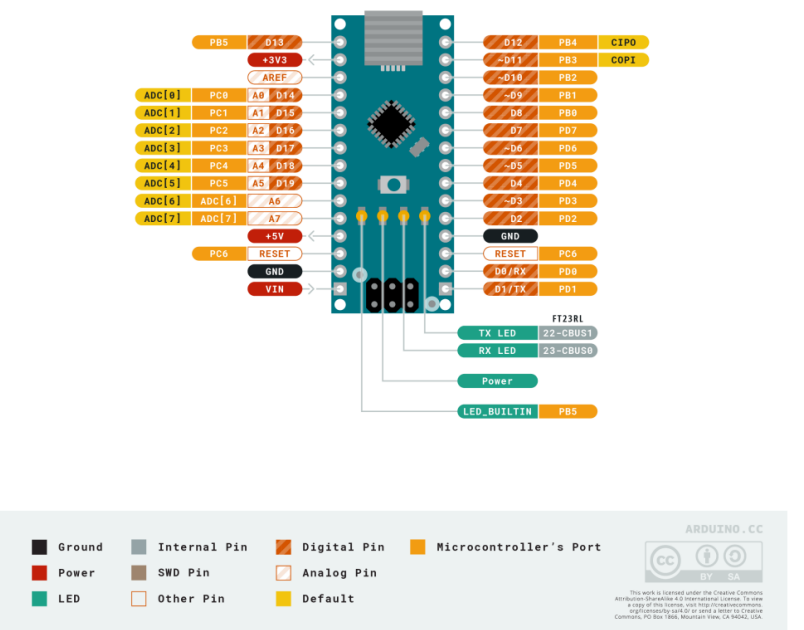
2. ไมโครคอนโทรลเลอร์ มีไมโครคอนโทรลเลอร์ภายใน ซึ่งเป็นสมองของ Arduino ที่ใช้ในการควบคุมและประมวลผลข้อมูล เช่น ATmega328P ในกรณีของ Arduino Nano ที่มาตรฐาน

3. ขาอินพุตและเอาต์พุต Arduino Nano มาพร้อมขาอินพุตและเอาต์พุตต่าง ๆ ที่ใช้ในการเชื่อมต่อกับอุปกรณ์ภายนอก เพื่อควบคุมและรับส่งข้อมูล

4. การเชื่อมต่อยูเอสบี มีพอร์ตยูเอสบี ที่ใช้ในการเชื่อมต่อ Arduino Nano กับคอมพิวเตอร์เพื่อโปรแกรมและส่งข้อมูล

5. ความหลากหลาย Arduino Nano มีความหลากหลายในรุ่นและเซ็นเซอร์ที่ใช้เพิ่มเติม เช่น Arduino Nano 33 อินเทอร์เน็ตประสานสรรพสิ่ง(Internet of Things, IoT) มีการเชื่อมต่อ Wi-Fi สำหรับโพรเจกต์อินเตอร์เน็ตสรรพสิ่ง

Arduino Nano เป็นเครื่องมือที่เหมาะสำหรับผู้เริ่มต้นและนักพัฒนาที่ต้องการทำโพรเจกต์ที่เกี่ยวกับการควบคุมอุปกรณ์และการสร้างอุปกรณ์อิเล็กทรอนิกส์ที่มีขนาดเล็กและเคลื่อนย้ายได้ง่าย



https://store.arduino.cc/products/arduino-nano: Accessed: May5, 2024.

รูปที่ 1.6 โครงสร้างในบอร์ด Arduino Nano

1.1.3 ตระกูล Arduino MKR

บอร์ดตระกูลนี้ มีคุณสมบัติเพื่อให้สามารถใช้สำหรับอินเทอร์เน็ตประสานสรรพสิ่งซึ่งบอร์ดในตระกูลนี้ประกอบไปด้วย

1.MKR 1000 Wi-Fi

2.MKR FOX 1200

3.MKR GSM 1400

4.MKR NB 1500

5.MKR Vidor 4000

6.MKR WAN 1300

7.MKR WAN 1310

8.MKR Wi-Fi 1010

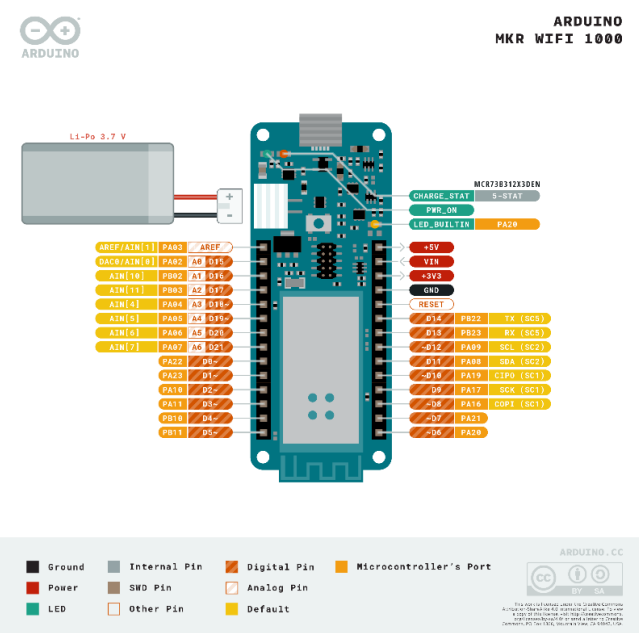
9.MKR Zero

ผู้เขียนขอแนะนำเฉพาะ MKR 1000 Wi-Fi ดังรูปที่ 1.7 เป็นทางเลือกที่ยอดเยี่ยมสำหรับผู้เริ่มต้น ผู้สร้าง หรือมืออาชีพในการเริ่มต้นด้วยเทคโนโลยีอินเทอร์เน็ตของสรรพสิ่ง ใช้หน่วยประมวลผล Arm® Cortex®-M0 32 บิต SAMD21 ที่เป็นที่นิยม และมี ECC508 crypto-chip เพื่อความปลอดภัย บอร์ดนี้เป็นส่วนหนึ่งของตระกูล MKR ที่ท่านสามารถเลือกจากชนิดหลายชนิดของชิลด์เพื่อสร้างโพรเจกต์โดยใช้บอร์ดนี้โดยไม่ต้องลงทุนมากในการเริ่มต้น

https://store.arduino.cc/products/arduino-mkr1000-wifi: Accessed: May5, 2024.

รูปที่ 1.7 บอร์ด Arduino MKR 1000 WiFi

บอร์ด Arduino MKR 1000 Wi-Fi เป็นบอร์ดไมโครคอนโทรลเลอร์ที่ออกแบบมาเพื่อการเชื่อมต่อกับเครือข่าย Wi-Fi และใช้ในโพรเจกต์ที่ต้องการการเชื่อมต่ออินเตอร์เน็ตในอุปกรณ์หรือโพรเจกต์ต่าง ๆ โดยใช้พลังงานต่ำ จากรูปที่ 1.8 สามารถอธิบายคุณสมบัติสำคัญได้ดังนี้



https://store.arduino.cc/products/arduino-mkr1000-wifi: Accessed: May5, 2024.

รูปที่ 1.8 โครงสร้างในบอร์ด Arduino MKR 1000 Wi-Fi

1. ไมโครคอนโทรลเลอร์ Arduino MKR 1000 Wi-Fi ใช้ไมโครคอนโทรลเลอร์หรือ MCU ประสิทธิภาพสูง ซึ่งมีการใช้งานต่อเนื่องในโหมดที่ต่ำพลังงาน ด้วยความเร็วนาฬิกาสูง
2. การเชื่อมต่อ Wi-Fi บอร์ดนี้มาพร้อมโมดูล Wi-Fi ภายในที่ใช้ในการเชื่อมต่อกับเครือข่าย Wi-Fi ในการสื่อสารแบบไร้สาย นี้ช่วยให้ผู้อ่านสามารถควบคุมอุปกรณ์หรือส่งข้อมูลผ่านอินเตอร์เน็ตได้อย่างง่าย
3. หน่วยความจำ มีหน่วยความจำเพียงพอสำหรับโปรแกรมและข้อมูลขนาดเล็ก ซึ่งเหมาะสำหรับโพรเจกต์ขนาดเล็กหรือกลาง
4. การต่อที่ครบ บอร์ด MKR 1000 Wi-Fi มาพร้อมหลายสายที่ครบครันเช่นหน่วยความจำ Micro SD, GPIO ดิจิทัลและอนาล็อก, พอร์ต ยูเอสบี, หัวต่อ I2C, UART, SPI, และอื่น ๆ ที่จำเป็นสำหรับการต่ออุปกรณ์เสริม
5. Arduino MKR 1000 Wi-Fi เป็นส่วนหนึ่งของระบบ Arduino ที่น่าเชื่อถือ มีการพัฒนาและสนับสนุนโดยชุมชนที่กว้างขวาง
6. การสนับสนุนการพัฒนาที่เป็นเลิศ: มี IDE (Integrated Development Environment) และไลบรารีอาร์ดูโนที่ช่วยให้ผู้อ่านสามารถเขียนโปรแกรมและโค้ดสำหรับบอร์ด MKR 1000 Wi-Fi ได้อย่างสะดวก
บอร์ด Arduino MKR 1000 Wi-Fi เหมาะสำหรับโพรเจกต์อินเทอร์เน็ตประสานสรรพสิ่งและโพรเจกต์ที่ต้องการการเชื่อมต่อ Wi-Fi ในโลกที่เชื่อมต่อกันในยุคนี้ เป็นเครื่องมือที่มีคุณค่าสำหรับนักพัฒนาและผู้สนใจในการสร้างอุปกรณ์อัจฉริยะและโพรเจกต์ต่าง ๆ ที่เชื่อมต่อกับอินเตอร์เน็ตไร้สายได้อย่างสะดวก

1.1.4 ตระกูลบอร์ด Arduino MEGA2560

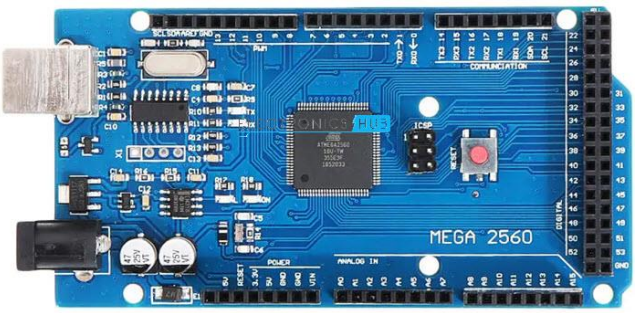
บอร์ดในตระกูลนี้ ถือว่าเป็นรุ่นใหญ่ เพราะว่าสามารถออกแบบให้ใช้ขาอินพุต/เอาต์พุตได้จำนวนมากกว่ารุ่นอื่นๆ ซึ่งในตระกูลนี้ จะประกอบไปบอร์ดดังต่อไปนี้

1.Due

2.GIGA R1 Wi-Fi

3.Mega 2560 Rev3

ผู้เขียนขอแนะนำบอร์ด Arduino Mega 2560 เป็นหนึ่งในรุ่นที่ใหญ่ที่สุดและมีความสามารถมากที่สุดในซีรีส์ Arduino ซึ่งออกแบบมาเพื่อให้สามารถใช้งานในโพรเจกต์ที่มีความซับซ้อนมากขึ้น แสดงได้ดังรูปที่ 1.9 และจากรูปที่ 1.10 จะแสดงคุณสมบัติและข้อมูลสำคัญเกี่ยวกับบอร์ด Arduino Mega 2560 ได้ดังนี้
1. ไมโครคอนโทรลเลอร์ Arduino Mega 2560 ใช้ไมโครคอนโทรลเลอร์ ATmega2560 ซึ่งเป็นไมโครคอนโทรลเลอร์ความเร็ว 16 MHz ที่มีความจุของแฟลชเมมโมรี่ 256 KB และ SRAM 8 KB ที่มากกว่า Arduino Uno ที่ทำให้มีความสามารถมากยิ่งขึ้นในการจัดการโค้ดและข้อมูล

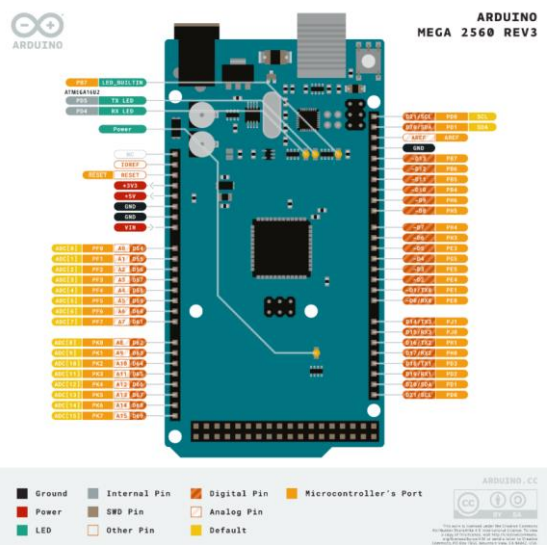


https://store.arduino.cc/products/arduino-mega-2560-rev3: Accessed: May5, 2024.

รูปที่ 1.9 บอร์ด Arduino MEGA2560

2. ขาอินพุต/เอาต์พุต Arduino Mega 2560 มีขาอินพุตและขาเอาต์พุตรวมทั้งหมด 54 ขา (digital I/O pins) โดยในนั้นมี 15 ขาเอาต์พุตที่สามารถใช้เป็น การปรับความกว้างของพัลส์ได้ นอกจากนี้ยังมีขาสำหรับการสื่อสารแบบ UART (Serial), I2C, และ SPI

3. การเชื่อมต่อยูเอสบี Arduino Mega 2560 มาพร้อมพอร์ตยูเอสบี สำหรับการเขียนโปรแกรมและอัปโหลดไฟล์ต่าง ๆ จากคอมพิวเตอร์ไปยังบอร์ด



https://store.arduino.cc/products/arduino-mega-2560-rev3: Accessed: May5, 2024.

รูปที่ 1.10 บอร์ด Arduino Mega2560

4. ความสามารถในการจัดเก็บไฟล์ Arduino Mega 2560 มีหน่วยความจำ (EEPROM) ขนาด 4 KB สำหรับการจัดเก็บข้อมูลและการจัดการการบันทึกข้อมูล
5. การจ่ายไฟ Arduino Mega 2560 สามารถใช้งานได้ในการจ่ายไฟในช่วง 7-12 โวลต์ (V) โดยใช้เส้นสายแบบดีซี หรือในบางกรณีสามารถใช้ยูเอสบีเป็นแหล่งจ่ายไฟ
6. สวิตช์เริ่มใหม่ บอร์ดมีสวิตช์ที่สามารถใช้ในการรีเซ็ตบอร์ด Arduino เพื่อให้โปรแกรมเริ่มทำงานใหม่
7. ส่วนประกอบ Arduino Mega 2560 สามารถใช้งานได้บนหลายระบบคอมพิวเตอร์รวมถึง Windows, Mac OS, และ Linux โดยมี Arduino IDE ที่รองรับหลายแพลตฟอร์ม
8. บอร์ดสำรองสำหรับแทนที่ Arduino Mega 2560 มีบอร์ดสำรองสำหรับแทนที่ ที่จะสามารถใส่ลงบนบอร์ดหลักเพื่อเพิ่มความสามารถและการเชื่อมต่ออุปกรณ์เสริมได้

Arduino Mega 2560 เหมาะสำหรับโพรเจกต์ที่มีความซับซ้อนมากขึ้น โพรเจกต์อินเทอร์เน็ตประสานสรรพสิ่ง การควบคุมอุปกรณ์และหุ่นยนต์ และโพรเจกต์ที่ต้องการมีขาอินพุตและเอาต์พุตจำนวนมาก บอร์ดนี้มีความสามารถมากที่สุดในซีรีส์ Arduino และเหมาะสำหรับนักพัฒนาที่มีประสบการณ์มากขึ้นในการทดลองและสร้างโพรเจกต์อิเล็กทรอนิกส์และการโปรแกรมที่ซับซ้อนมากขึ้น

สรุปท้ายบท

ทั้งหมดของบอร์ด Arduino แต่ละรุ่นเสนอคุณสมบัติและความสามารถที่แตกต่างกัน แต่จะสรุปรวมท้ายบท เกี่ยวกับบอร์ด Arduino ดังนี้

1. ความหลากหลายมีหลายรุ่นของบอร์ด Arduino ที่เหมาะกับความต้องการและโพรเจกต์ที่แตกต่างกัน เช่น Arduino Uno, Arduino Mega, Arduino Due, และอื่น ๆ
2. การต่อเพิ่มอุปกรณ์ บอร์ด Arduino เพื่อการต่อเพิ่มอุปกรณ์เสริม และชุดคำสั่งที่เหมาะสำหรับการต่อใช้งานกับอุปกรณ์เสริม
3. ราคาที่ย่อมเยา บอร์ด Arduino มีราคาที่ย่อมเยา และเหมาะสำหรับนักพัฒนาทุกคน ไม่ว่าจะเป็นนักศึกษา นักเรียน หรือนักพัฒนาระดับอาชีพ
4. มีชุมชนและการสนับสนุน Arduino ที่กว้างขวางและการสนับสนุนจากผู้พัฒนาทั่วโลก ที่ช่วยแก้ปัญหาและให้คำปรึกษาได้
5. สมรรถนะของบอร์ด Arduino มีคุณสมบัติที่ดี, ที่ช่วยให้ผู้อ่านเข้าใจและใช้งานได้อย่างสะดวก

บอร์ด Arduino มีความยืดหยุ่นและเหมาะสำหรับผู้ที่สนใจในการเริ่มต้นด้วยการพัฒนาอุปกรณ์อิเล็กทรอนิกส์และโพรเจกต์ที่เชื่อมต่อกับโลกดิจิทัลและอินเทอร์เน็ตของสรรพสิ่งได้สะดวก

คำถามท้ายบท

1. บอร์ด Arduino Uno ใช้ไมโครคอนโทรลเลอร์รุ่นอะไร

2. บอร์ด Arduino รุ่นไหนที่ใช้ Wi-Fi ได้

3. บอร์ด Arduino Mega 2560 มีจำนวนพินดิจิตอล I/O ทั้งหมดกี่พิน

4. บอร์ด Arduino Uno ใช้ไมโครคอนโทรลเลอร์รุ่นอะไร

5. บอร์ด Arduino Nano 33 BLE มีการเชื่อมต่อไร้สายแบบใด

เอกสารอ้างอิง

1. เดชฤทธิ์ มณีธรรม, *คัมภีร์การใช้งานไมโครคอนโทรลเลอร์ Arduino*, กรุงเทพฯ: ซีเอ็ดยูเคชั่น, 2560.

2. สุวิทย์ เมาะราษี, *การโปรแกรมมิ่งบอร์ด Arduino ด้วย C# .NET และ Sketch*, สำนักพัฒนาสมรรถนะครูและบุคลากรอาชีวศึกษา, 2563.

3. "*Arduino - Home*," Arduino.cc. [Online]. Available: https://www.arduino.cc/., 2023

4. S. Fitzgerald and M. Shiloh, *The Arduino Projects Book*, Torino, Italy: Arduino, 2012.

5. https://store.arduino.cc/products/arduino-uno-rev3: Accessed: May5, 2024.

6. https://store.arduino.cc/products/arduino-uno-wifi-rev2: Accessed: May5, 2024.

7. https://store.arduino.cc/products/arduino-nano: Accessed: May5, 2024.

8. https://store.arduino.cc/products/arduino-mega-2560-rev3: Accessed: May5, 2024.

บทที่ 2 การเขียนโปรแกรมควบคุมบอร์ด Arduino

ในบทที่1 มีการกล่าวถึงบอร์ด Arduino ในแต่ละตระกูล ซึ่งเป็นส่วนของฮาร์ดแวร์ ในบทนี้จะเป็นการแนะนำซอฟท์แวร์ที่ใช้ในการเขียนโปรแกรมเพื่อใช้ควบคุมบอร์ด Arduino ซึ่งโปรแกรมที่ใช้จะมี 3 โปรแกรม ดังต่อไปนี้ คือ

2.1 โปรแกรม Visual Studio

โปรแกรม Visual Studio เป็นสภาพแวดล้อมพัฒนาซอฟต์แวร์ (Integrated Development Environment, IDE) ที่พัฒนาโดย Microsoft สำหรับการพัฒนาซอฟต์แวร์และแอปพลิเคชันต่าง ๆ โดยเฉพาะเพื่อพัฒนาแอปพลิเคชันบนระบบปฏิบัติการ Windows และระบบฐานข้อมูลต่าง ๆ เป็นเครื่องมือที่หลากหลายและมีความสามารถในการสนับสนุนการพัฒนาซอฟต์แวร์ที่หลากหลาย รวมถึงแอปพลิเคชันเบื้องต้นและการพัฒนาโปรแกรมคอมพิวเตอร์ที่ซับซ้อนมากขึ้นด้วย Visual Studio มีรุ่นและเวอร์ชันต่าง ๆ ที่เหมาะสำหรับการพัฒนาแอปพลิเคชันที่แตกต่างกัน ดังนี้

1.1 Visual Studio Community เวอร์ชันฟรีสำหรับนักพัฒนาทั่วไป ใช้สำหรับพัฒนาแอปพลิเคชันบน Windows, Android, iOS, web, และคอมพิวเตอร์ต่าง ๆ มีความสามารถในการพัฒนาทั้งแอปพลิเคชันเบื้องต้นและคอมพิวเตอร์ซับซ้อน

1.2 Visual Studio Professional เวอร์ชันที่มีความสามารถและเครื่องมือเพิ่มเติมสำหรับนักพัฒนามืออาชีพ มีการสนับสนุนในการพัฒนาแอปพลิเคชันองค์กรและโพรเจกต์ที่มากขึ้น

1.3 Visual Studio Enterprise เวอร์ชันที่มีความสามารถและเครื่องมือที่มากขึ้นสำหรับการพัฒนาโพรเจกต์แอปพลิเคชันที่มีขนาดใหญ่และซับซ้อน รวมถึงเครื่องมือในการทดสอบและคุณสมบัติคุณภาพ

Visual Studio ช่วยนักพัฒนาในการเขียนโปรแกรม, การทดสอบ, การบริหารโพรเจกต์, และการพัฒนาแอปพลิเคชันอย่างมีประสิทธิภาพ โดยมีเครื่องมือและคุณสมบัติที่สมรรถภาพสูงสำหรับการพัฒนาซอฟต์แวร์ ซึ่งผู้อ่านสามารถใช้ Visual Studio Community มาใช้งานฟรีได้ ไม่เสียค่าซอฟท์แวร์ (สามารถดาวน์โหลดที่ https://visualstudio.microsoft.com/vs/community/)

2.2 Arduino Web Editor

เป็นบริการออนไลน์ที่พัฒนาขึ้นโดย Arduino ซึ่งเป็นบริษัทที่มีชื่อเสียงในการสร้างและพัฒนาบอร์ด Arduino และซอฟต์แวร์ที่เกี่ยวข้อง เป้าหมายหลักของ Arduino Web Editor คือการอำนวยความสะดวกให้กับผู้ใช้ Arduino ในการเขียนโปรแกรมผ่านอินเตอร์เน็ต โดยไม่จำเป็นต้องติดตั้งซอฟต์แวร์ออกไปยังเครื่องคอมพิวเตอร์ นี่คือการอธิบายขั้นตอนพื้นฐานของ Arduino Web Editor

1. การสมัครสมาชิก เพื่อใช้งาน Arduino Web Editor ผู้ใช้จำเป็นต้องลงทะเบียนและสร้างบัญชีผู้ใช้บน Arduino เว็บไซต์

2. การเข้าสู่ระบบ หลังจากสมัครสมาชิกเสร็จแล้ว ผู้ใช้สามารถเข้าสู่ระบบโดยใช้ชื่อผู้ใช้และรหัสผ่านของตน
3. การเลือกบอร์ด ผู้ใช้สามารถเลือกบอร์ด Arduino ที่ต้องการโปรแกรมจากเมนูที่มีให้เลือก บอร์ดที่รองรับอยู่ใน Arduino Web Editor มี Arduino Uno, Arduino Mega, Arduino Nano, และอื่น ๆ
4. การเขียนโปรแกรม ผู้ใช้สามารถเขียนโปรแกรม Arduino ของตนเป็นภาษา C/C++ โดยใช้เครื่องมือการเขียนโปรแกรมออนไลน์ที่มีให้ใน Arduino Web Editor มีอุปกรณ์ช่วยเหลือในการเขียนโปรแกรมเช่นเดียวกับการโปรแกรมบอร์ดในแฮร์ดแวร์ Arduino ปกติ
5. การอัปโหลดและโปรแกรม ผู้ใช้สามารถอัปโหลดโปรแกรมที่เขียนเสร็จแล้วลงบอร์ด Arduino ของพวกเขาผ่านเครื่องมือออนไลน์ โดยทั่วไปแล้ว, Arduino Web Editor จะเชื่อมต่อกับบอร์ดผ่านทางพอร์ตยูเอสบี และส่งโปรแกรมไปยังบอร์ดเพื่อการโปรแกรม
6. การบันทึกและโปรแกรม ผู้ใช้สามารถบันทึกโปรแกรมของพวกเขาในออนไลน์และเข้าถึงโปรแกรมของพวกเขาจากทุกที่ที่มีการเชื่อมต่ออินเตอร์เน็ต

Arduino Web Editor เป็นเครื่องมือที่เหมาะสำหรับผู้ใช้ที่ต้องการเขียนและโปรแกรม Arduino แต่ไม่ต้องการติดตั้งซอฟต์แวร์บนคอมพิวเตอร์ของพวกเขา มีอินเทอร์เฟซออนไลน์ที่ใช้งานง่ายและใช้งานได้ทุกที่ที่มีการเชื่อมต่ออินเตอร์เน็ต แต่อย่างไรก็ตาม เพื่อให้ง่ายต่อการใช้งาน ผู้เขียนจึงอธิบายการใช้งาน Arduino IDE แบบลงโปรแกรมที่เครื่องคอมพิวเตอร์ที่ต้องการติดตั้งดังหัวข้อถัดไป เป็นหลัก

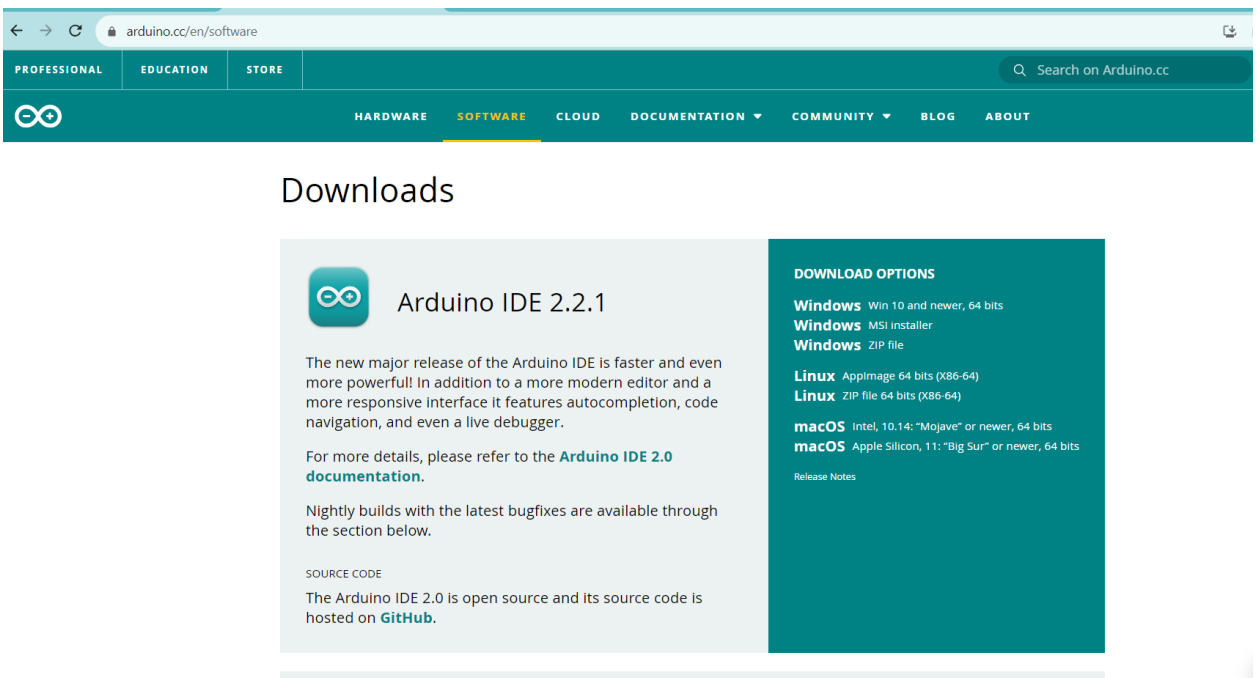
2.3 โปรแกรม Arduino IDE

Arduino IDE (Integrated Development Environment) คือโปรแกรมคอมพิวเตอร์ที่สร้างขึ้นโดย Arduino.cc ซึ่งมีลักษณะคล้ายกันกับ ภาษา C/C^{++} เพื่อใช้ในการพัฒนาและโปรแกรมบอร์ด Arduino และอุปกรณ์อิเล็กทรอนิกส์ที่ใช้ไมโครคอนโทรลเลอร์ Arduino โดยสำเร็จรูป เวอร์ชัน IDE นี้เป็นซอฟต์แวร์ที่เปิดตัวและฟรีให้ใช้งานแก่นักพัฒนาทุกคน Arduino IDE มีคุณสมบัติและเครื่องมือที่สมรรถภาพสูงสำหรับการพัฒนาโปรแกรมองค์กรอย่างง่าย นักพัฒนาสามารถเขียนโปรแกรมโดยใช้ภาษาโปรแกรมอาร์ดูโนออนที่เข้าใจง่ายและเหมาะสำหรับผู้ที่ไม่มีความรู้มากในการเขียนโปรแกรม นอกจากนี้ มีเครื่องมือในการอัปโหลดโปรแกรมลงบอร์ด Arduino ติดตามข้อผิดพลาดในโปรแกรม และโหลดไลบรารีอาร์ดูโนออนเพิ่มเติมเพื่อเพิ่มความสามารถในการพัฒนา ซึ่งในหนังสือเล่มนี้ ผู้เขียนจะใช้โปรแกรมควบคุมฮาร์ดแวร์ผ่าน Arduino ด้วยโปรแกรม Arduino IDE เป็นหลัก และจะอธิบายการติดตั้ง และเขียนโปรแกรม Arduino IDE ดังหัวข้อต่อไปนี้

2.3.1การติดตั้ง และใช้งาน Arduino IDE V2

โปรแกรม Arduino IDE เป็นเครื่องมือที่ใช้เขียนโปรแกรมลงบอร์ดใน Arduino โดยมีส่วนเขียนโปรแกรม (Editor) ส่วนแปลโปรแกรม ส่วนอัปโหลดโปรแกรม ส่วนเลือกบอร์ด-พอร์ต ส่วนติดตั้งไลบรารี ส่วนเปิดโปรแกรมตัวอย่าง ส่วนติดตั้งบอร์ดใหม่ ส่วน Serial Monitor ส่วน Serial Plotter มีฟีเจอร์จัดรูปแบบโปรแกรมโปรแกรมอัตโนมัติ สำหรับ Arduino IDE V2 มีฟีเจอร์ Autocomplete และเลือกธีมของโปรแกรม

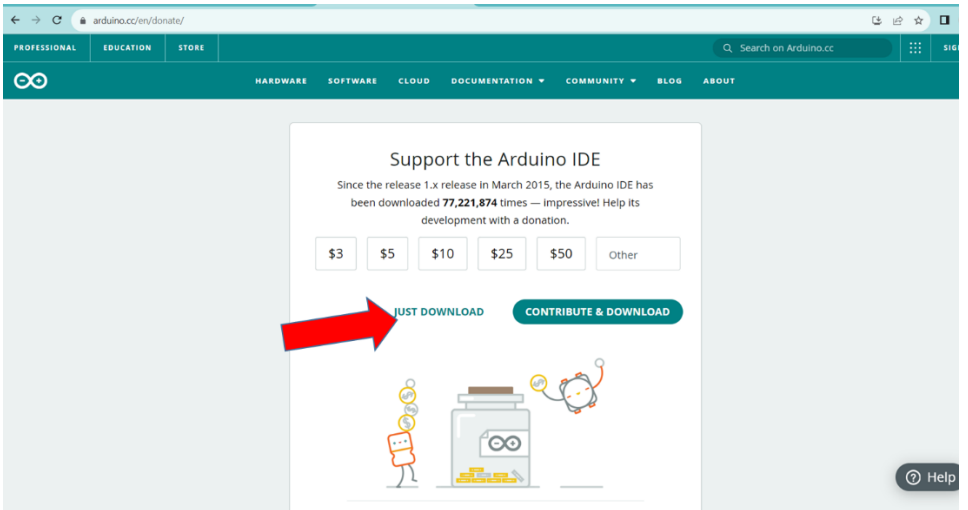
เพิ่มขึ้นมา Arduino IDE V1 และ Arduino IDE V2 มี UI/UX ที่แตกต่างกัน เนื่องจาก Arduino IDE V2 ทำขึ้นมาใหม่ทั้งหมด ไม่ได้พัฒนามาจาก Arduino IDE V1 มีการแนะนำการใช้งาน Arduino IDE V2 เบื้องต้น ติดตั้งโปรแกรม ทดลองติดตั้งบอร์ด ESP32 ติดตั้งไลบรารี ใช้งาน Serial Monitor และ Serial Plotter เข้าไปที่ https://www.arduino.cc/en/software ที่ส่วน Downloads ให้เลือกดาวน์โหลดตรงตามรุ่นของระบบปฏิบัติการคอมพิวเตอร์ ตัวอย่างใช้ Windows 10 64 บิต จึงเลือกดาวน์โหลด Windows ดังรูปที่ 2.1



https://www.arduino.cc/en/software: Accessed:Oct.23, 2023.

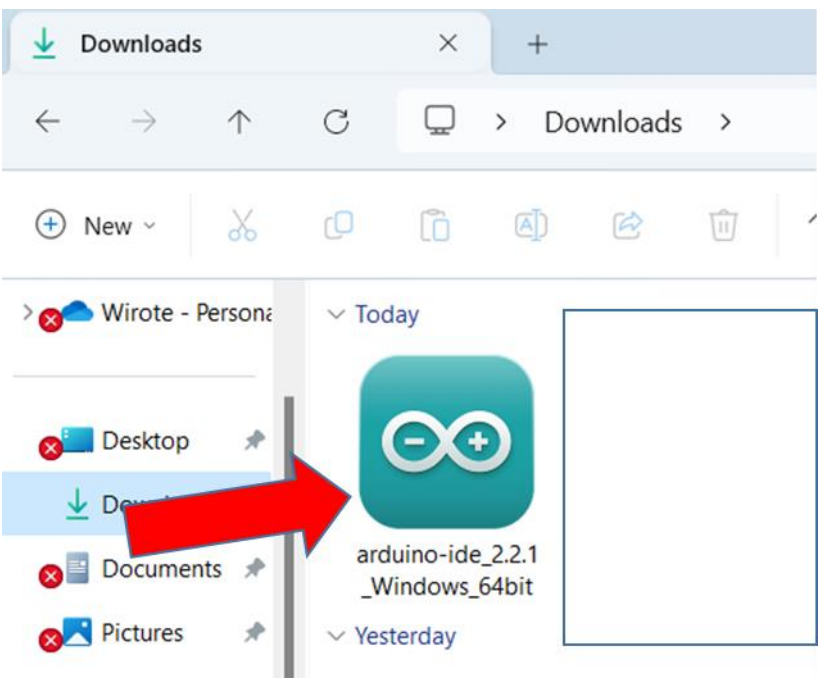
รูปที่ 2.1 หน้าเมนูในการดาวน์โหลดโปรแกรม

จะขึ้นหน้าแนะนำให้บริจาคดังรูปที่ 2.2 หากยังไม่สะดวกให้กดปุ่ม JUST DOWNLOAD



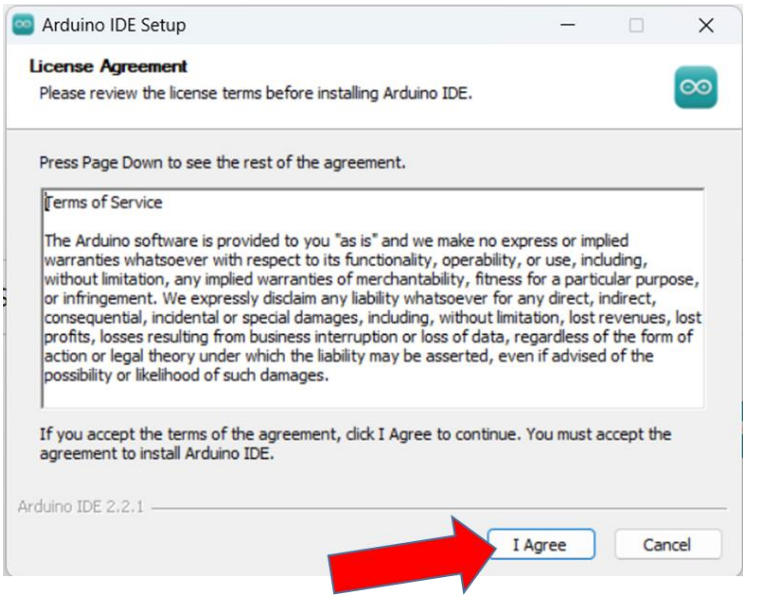
รูปที่ 2.2 เมนูในการเชิญชวนบริจาคเงินก่อนดาวน์โหลด

รอจนกว่าจะดาวน์โหลดเสร็จสิ้น ได้ไฟล์ติดตั้งชื่อ arduino-ide_x.x.x_Windows_64bit.exe มาดังรูปที่ 2.3



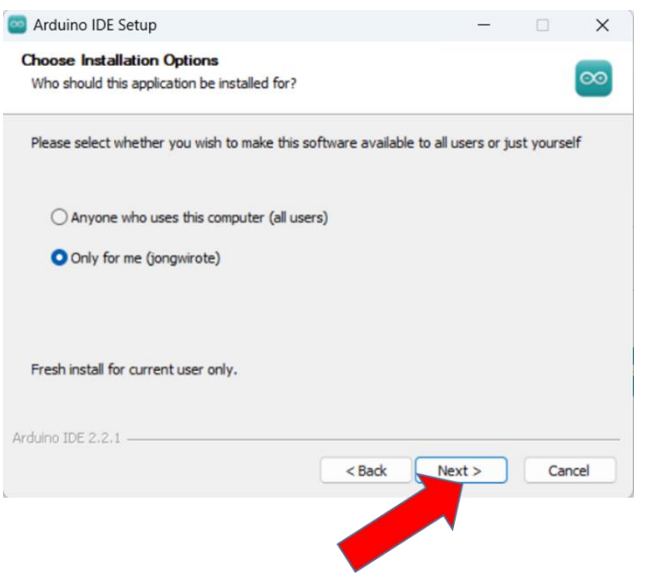
รูปที่ 2.3 ไอคอนของโปรแกรมที่โหลดมาได้

ดับเบิลคลิกเปิดไฟล์ติดตั้ง ในหน้าแรกของการติดตั้งจะขึ้นเงื่อนไขการใช้งานโปรแกรมมาดังรูปที่ 2.4 ให้กดปุ่ม I Agree เพื่อยอมรับเงื่อนไขการใช้งาน



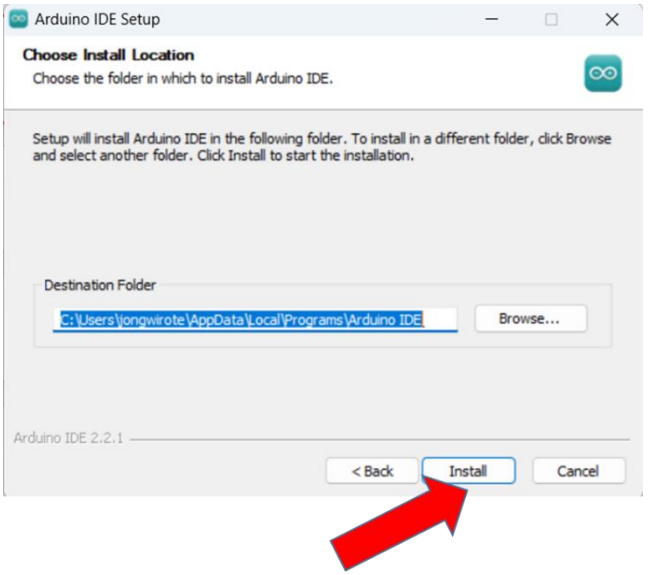
รูปที่ 2.4 เมนูในการยอมรับข้อตกลงก่อนติดตั้งโปรแกรม

จากนั้นหน้าต่างติดตั้งจะถามว่า ให้ติดตั้งโปรแกรมนี้ให้กับทุก User ในคอมพิวเตอร์เครื่องนี้ หรือเฉพาะ User นี้ดังรูปที่ 2.5 แนะนำให้ใช้ Only for me แล้วกดปุ่ม Next



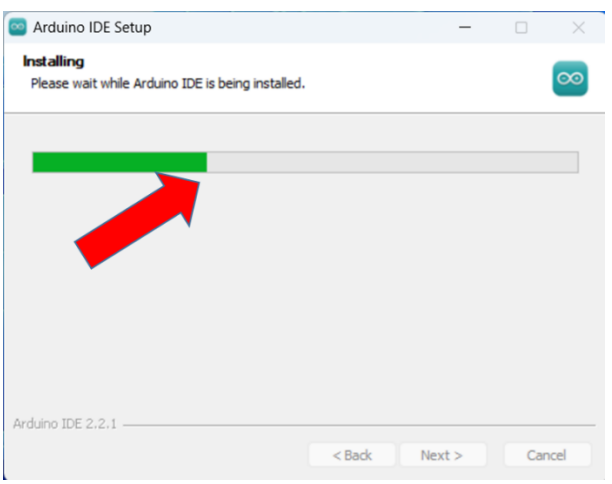
รูปที่ 2.5 เมนูในการเลือกว่าจะให้ทุกคนใช้หรือเฉพาะคุณ

จากนั้นหน้าต่างติดตั้งจะให้กำหนดโฟลเดอร์ที่เก็บโปรแกรมดังรูปที่ 2.6 แนะนำให้ใช้โฟลเดอร์เดิม กดปุ่ม Install เพื่อติดตั้งทันที



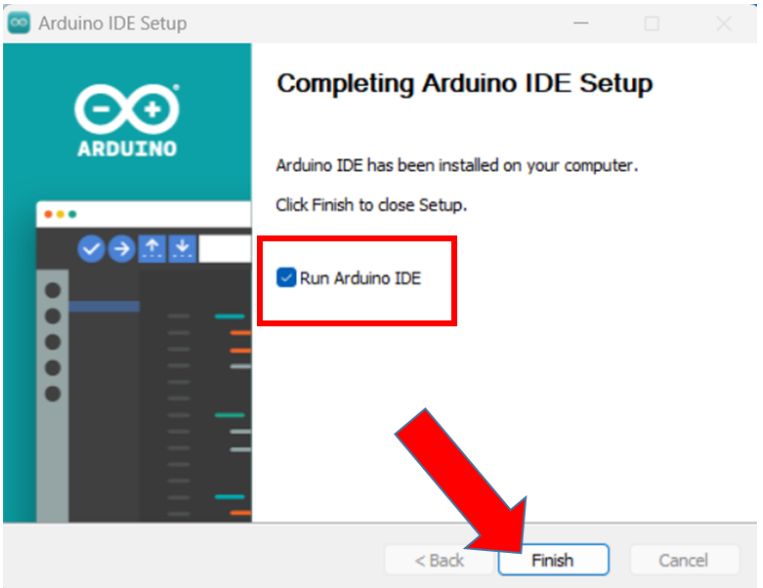
รูปที่ 2.6 โฟลเดอร์ที่ติดตั้งโปรแกรม

รอจนกว่าโปรแกรมจะติดตั้งเสร็จดังรูปที่ 2.7



รูปที่ 2.7 ให้เห็นโปรแกรมขณะกำลังติดตั้ง

เมื่อติดตั้งเสร็จ หน้าต่างติดตั้งจะแสดงดังรูปที่ 2.8 หากต้องการเปิดโปรแกรม Arduino IDE ทันที ให้เลือก Run Arduino IDE จากนั้นกดปุ่ม Finish

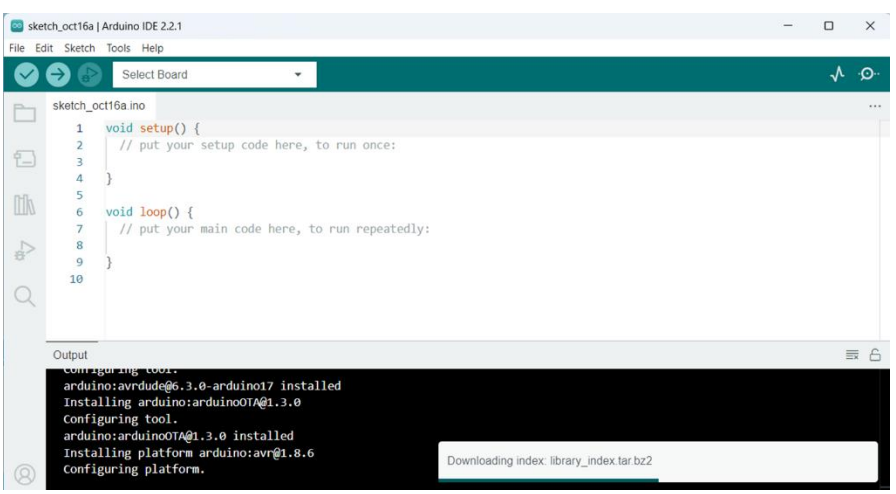


รูปที่ 2.8 เมนูผลลัพธ์เมื่อติดตั้งโปรแกรมสำเร็จ

ที่หน้าเดสทอป จะมีไอคอนโปรแกรม Arduino IDE ดังรูปที่ 2.9 เพิ่มขึ้นมาแล้ว ดับเบิลคลิกที่ไอคอนเพื่อเปิดโปรแกรมทันที

รูปที่ 2.9 ไอคอนของโปรแกรมที่ติดตั้งเสร็จ

โปรแกรม Arduino IDE V2 จะเปิดขึ้นมา ดังรูปที่ 2.10



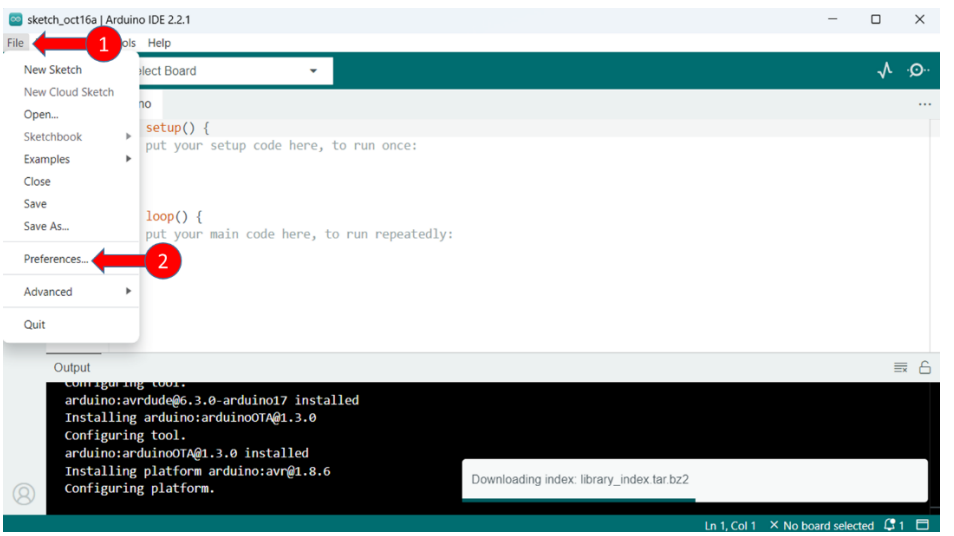
รูปที่ 2.10 หน้าเมนูเมื่อมีการเปิดใช้โปรแกรม

2.3.2 การตั้งค่าโปรแกรม

กดเมนู File เลือก Preferences...ดังรูปที่ 2.11

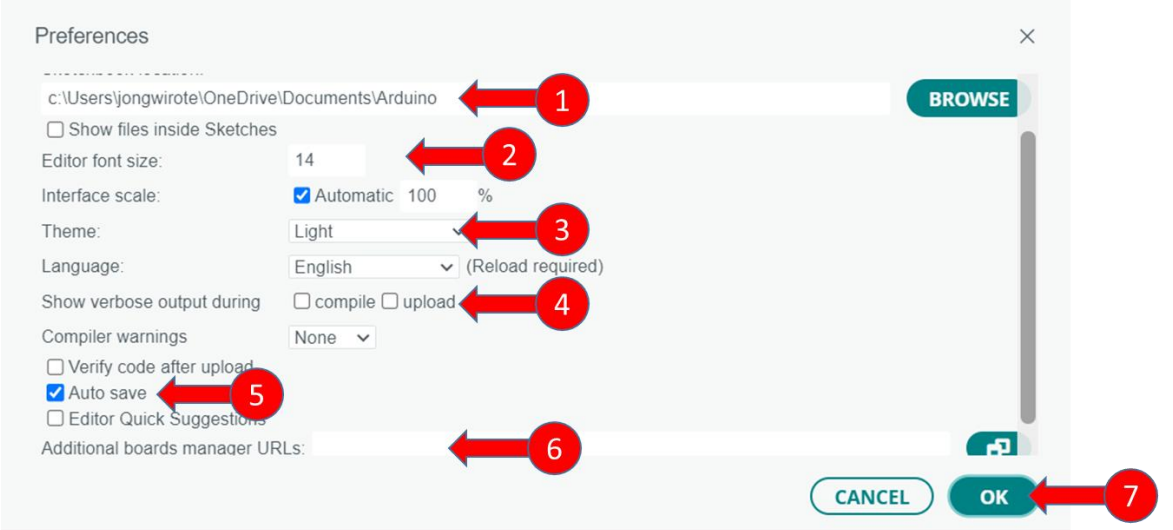
ในหน้าต่าง Preferences มีตัวเลือกให้ตั้งค่าดังรูปที่ 2.12 มีรายละเอียดดังนี้

1. โฟลเดอร์เก็บโปรแกรมโปรแกรมหลัก และเก็บไลบรารี
2. ขนาดตัวอักษร ส่วนเขียนโปรแกรม
3. ธีม ของโปรแกรม สามารถกดเปลี่ยนได้ตามต้องการ
4. ให้แสดงผลขั้นตอนการทำงานแบบละเอียดในขั้นตอนไหนบ้าง แนะนำให้กดเลือกทั้ง compile และ upload



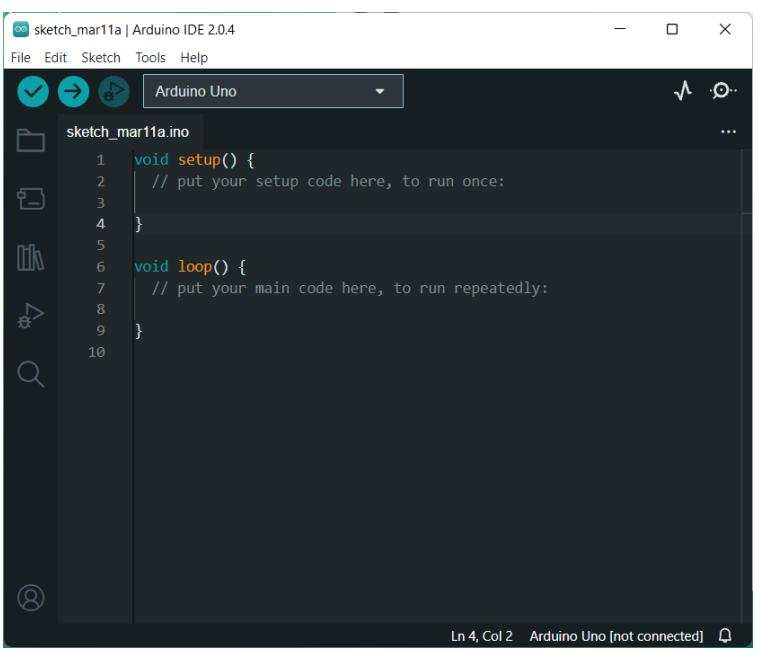
รูปที่ 2.11 เมนูในการเลือก Preferences

5. เลือกให้บันทึกไฟล์อัตโนมัติเมื่อแปลโปรแกรม หรืออัปโหลดโปรแกรม (แนะนำ)
6. ช่องใส่เชื่อมโยงสำหรับติดตั้งบอร์ดเพิ่ม (รายละเอียดในหัวข้อถัดไป)
7. เมื่อตั้งค่าเสร็จแล้ว กดปุ่ม OK เพื่อบันทึก



รูปที่ 2.12 เมนูย่อยที่ใช้ในการเลือกค่าใน Preferences

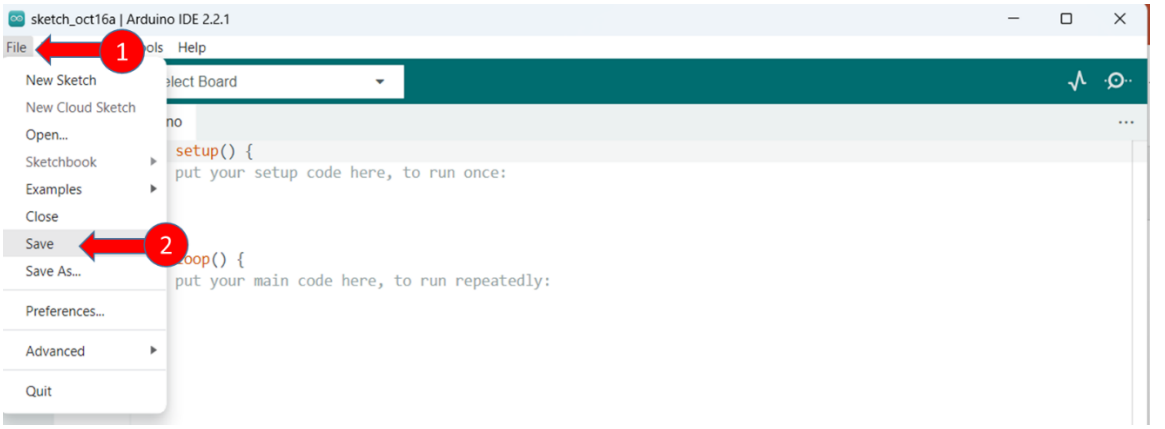
ตัวอย่างกดเปลี่ยนธีมเป็น Dark หน้าตาโปรแกรมเปลี่ยนไปดังรูปที่ 2.13



รูปที่ 2.13 หน้าเมนูเมื่อเปิดใช้โปรแกรมแบบ Dark mode

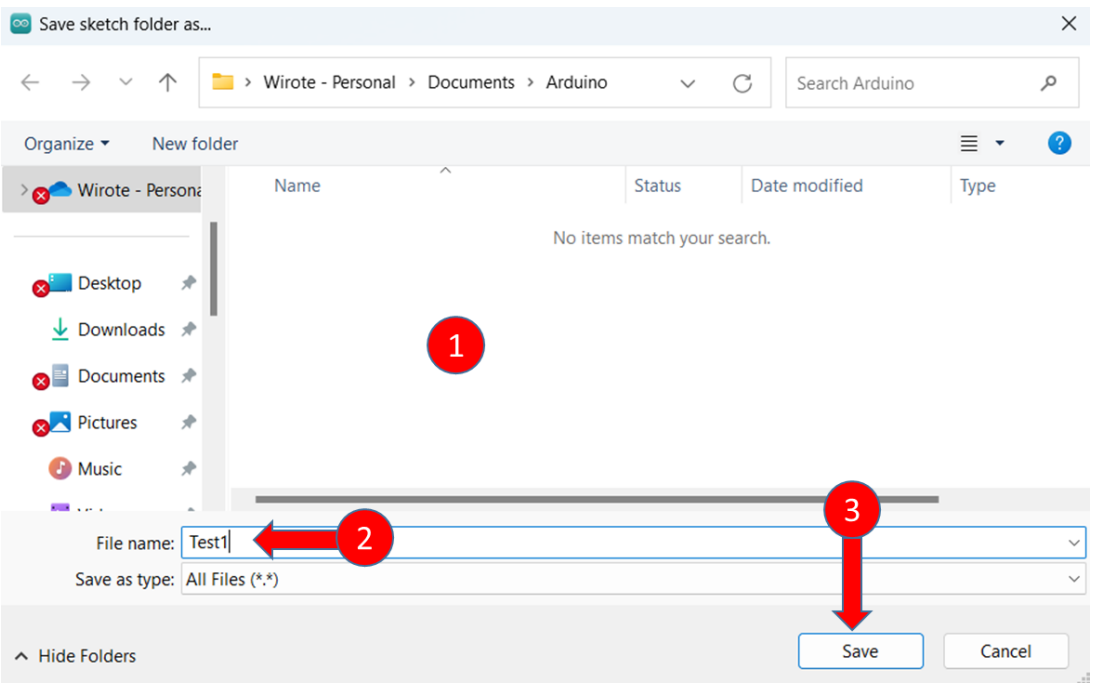
2.3.3 การบันทึกไฟล์ - เปิดไฟล์

การบันทึกไฟล์โปรแกรม ทำได้โดยกดเมนู File เลือก Save หรือกดปุ่ม Ctrl + S บนคีย์บอร์ดดังรูปที่ 2.14



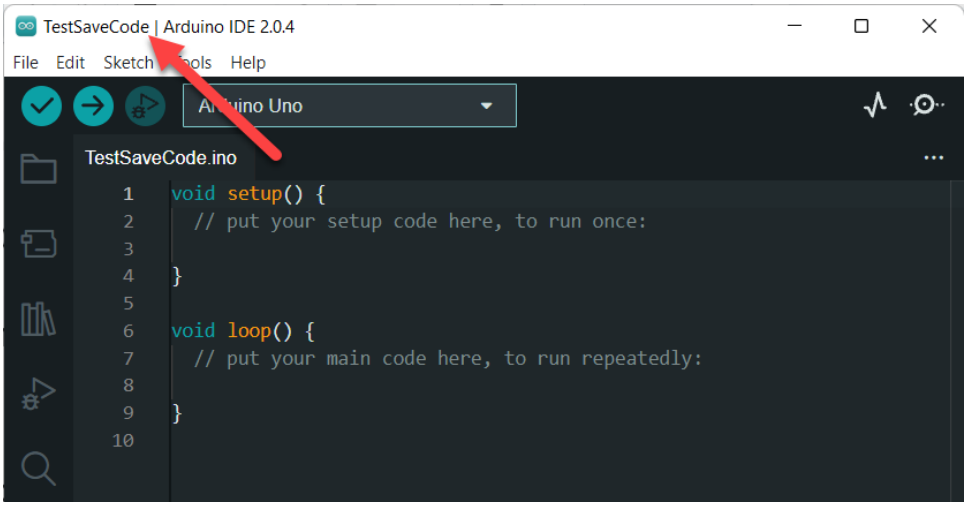
รูปที่ 2.14 เมนูในการบันทึกโปรแกรมที่เขียนขึ้น

เลือกโฟลเดอร์เก็บโปรแกรม (1) ตั้งชื่อ (2) แล้วกดปุ่ม Save (3) ดังรูปที่ 2.15



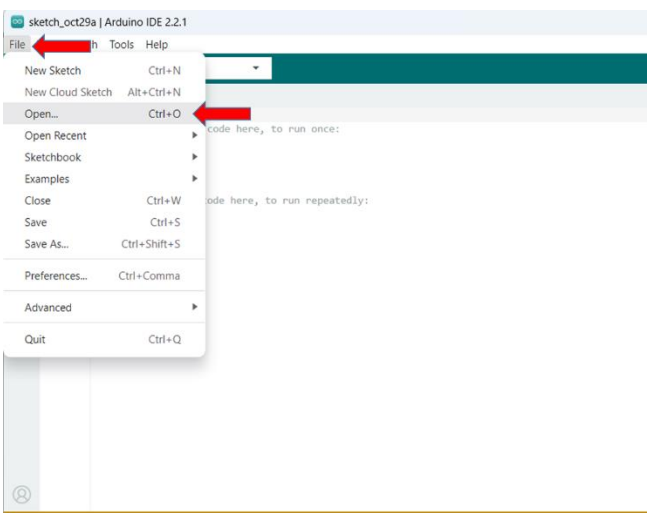
รูปที่ 2.15 เมนูสำหรับตั้งชื่อไฟล์ที่จะบันทึก

เมื่อบันทึกเรียบร้อย ชื่อที่บันทึกจะแสดงในส่วนหัวของโปรแกรมดังรูปที่ 2.16



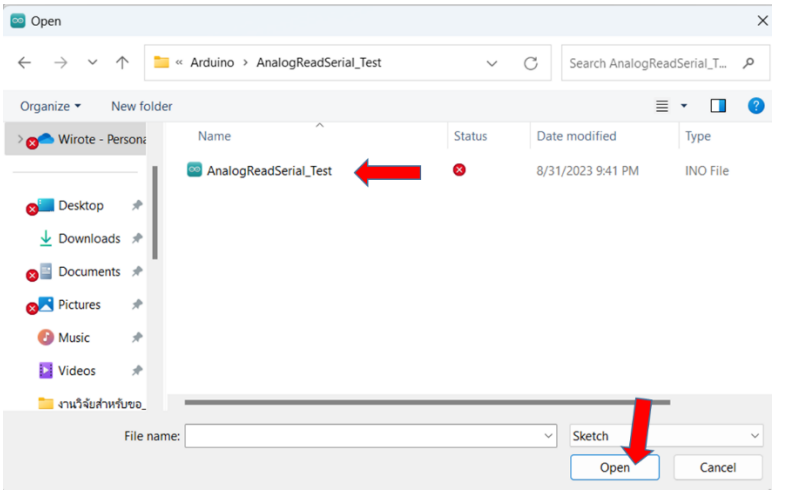
รูปที่ 2.16 การเปิดโปรแกรมจากชื่อไฟล์ที่เลือกไว้

การเปิดโปรแกรมทำได้โดยกด File > Open ดังรูปที่ 2.17



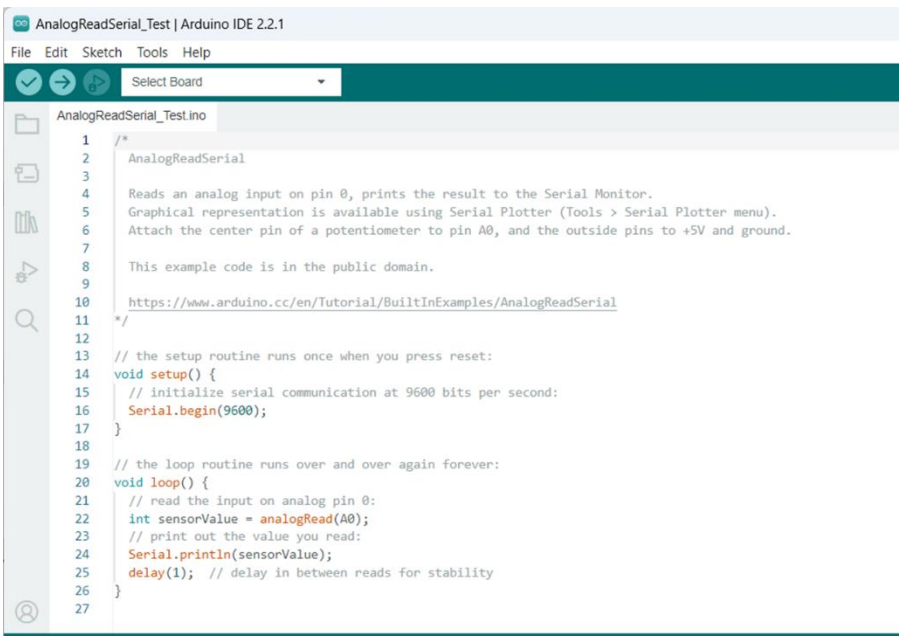
รูปที่ 2.17 เมนูสำหรับการเลือกเปิดไฟล์โปรแกรมที่บันทึกไว้

เลือกไฟล์ .ino แล้วกดปุ่ม Open ดังรูปที่ 2.18



รูปที่ 2.18 วิธีการเลือกไฟล์โปรแกรมที่ต้องการเปิดใช้งาน

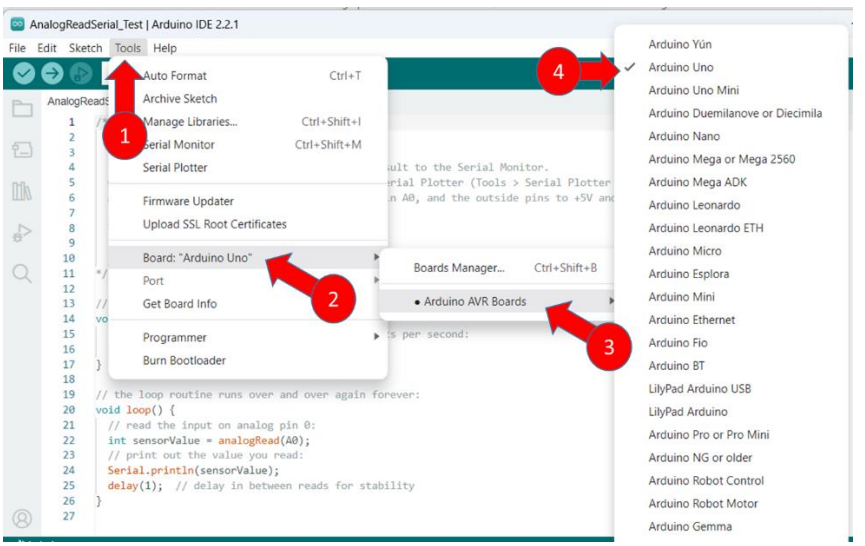
หน้าต่าง Arduino IDE จะเปิดขึ้นมาใหม่ และมีโปรแกรมแสดงขึ้นมา ดังรูปที่ 2.19



รูปที่ 2.19 ผลลัพธ์เมื่อเปิดไฟล์โปรแกรมตามที่เลือกไว้

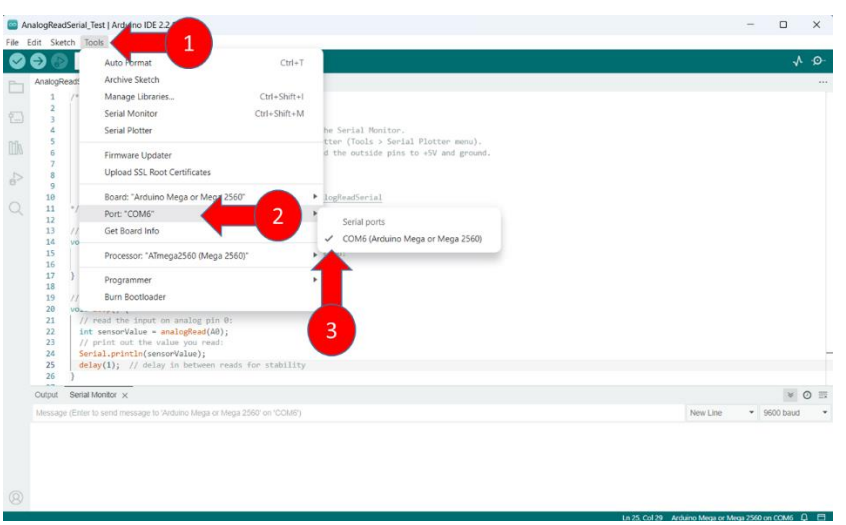
2.3.4 การเลือกบอร์ด เลือกพอร์ต

ก่อนอัปโหลดโปรแกรม ต้องเลือกบอร์ด และพอร์ตก่อน การเลือกบอร์ดทำได้โดยกดเมนู Tool > Board แล้วเลือกบอร์ดที่ต้องการ ตัวอย่าง ต้องการเลือกบอร์ด Arduino Uno อยู่ในกลุ่ม Arduino AVR Boards แล้วกด Arduino Uno ดังรูปที่ 2.20



รูปที่ 2.20 วิธีการเลือกบอร์ด Arduino ให้ตรงกับรุ่นที่ใช้งาน

การเลือกพอร์ต ต้องต่อบอร์ดเข้ากับเครื่องคอมพิวเตอร์ก่อน จากนั้นกด Tool > Port แล้วเลือกพอร์ตที่แสดง ดังรูปที่ 2.21 หากแสดงหลายพอร์ต ให้ค้นหาพอร์ตได้จากบทความ การหาพอร์ตของบอร์ด Arduino NodeMCU, ESP32, ESP8266



รูปที่ 2.21 วิธีการเลือกพอร์ตที่เชื่อมกับบอร์ด Arduino

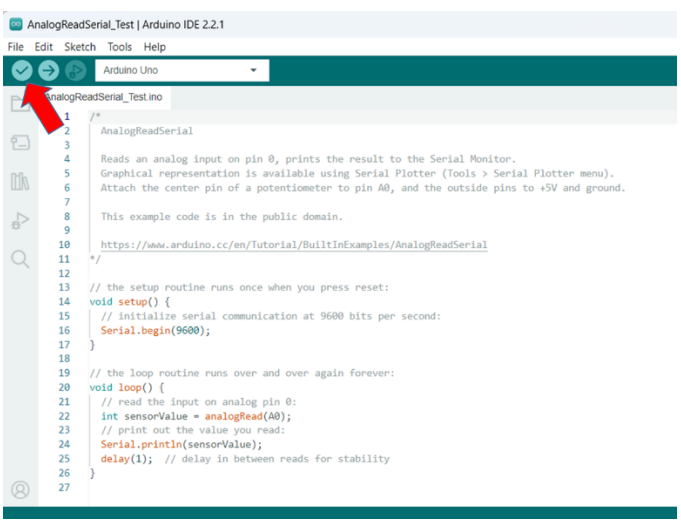
บอร์ดและพอร์ตที่เลือกจะแสดงที่มุมขวาล่างของหน้าต่างโปรแกรมดังรูปที่ 2.22

รูปที่ 2.22 ผลลัพธ์เมื่อเลือกบอร์ดและพอร์ตบนเมนูหน้าต่างโปรแกรม

2.3.5 การแปลโปรแกรม - อัปโหลดโปรแกรม

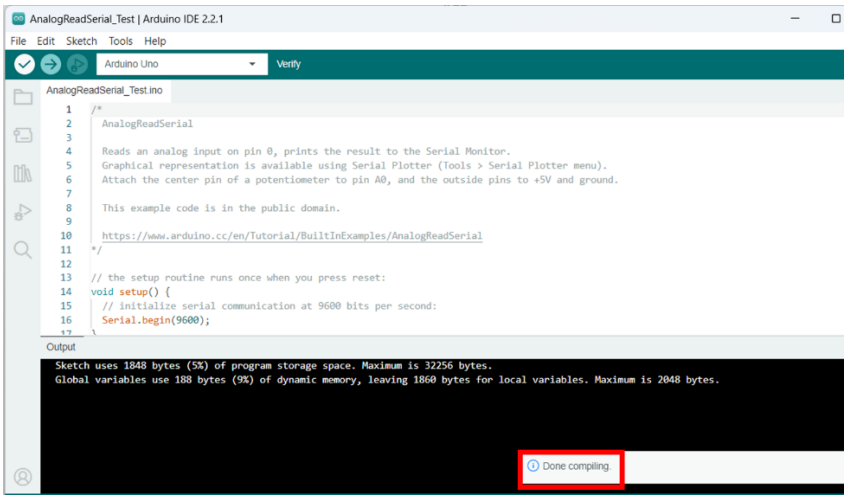
การแปลโปรแกรม คือการแปลงโปรแกรมโปรแกรมภาษา C/C++ ให้เป็น Machine Code (โปรแกรมที่ CPU เข้าใจ) โดยมักใช้ตรวจสอบว่าโปรแกรมมีข้อผิดพลาดหรือไม่ ก่อนอัปโหลดโปรแกรม

การแปลโปรแกรม ทำได้โดยกดปุ่ม Verify ดังรูปที่ 2.23



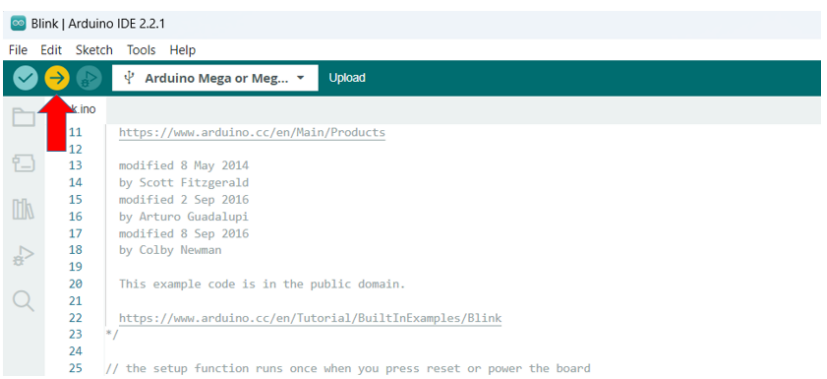
รูปที่ 2.23 วิธีการเลือกแปลไฟล์โปรแกรมที่ใช้งาน

ในระหว่างแปลโปรแกรม จะมีข้อความวิ่งในช่อง Console และหากเขียนโปรแกรมถูกต้อง ไม่มี Error จะมีข้อความ Done compiling แสดงดังรูปที่ 2.24



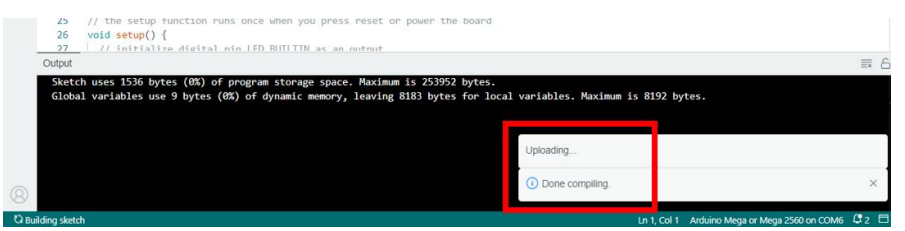
รูปที่ 2.24 ผลลัพธ์เมื่อแปลไฟล์โปรแกรมสำเร็จ

การอัปโหลดโปรแกรม จะต้องเลือกบอร์ด และพอร์ตก่อน (หากยังไม่ได้เลือก) จากนั้นกดปุ่มอัปโหลด ดังรูปที่ 2.25



รูปที่ 2.25 เมนูในการเลือกอัปโหลดโปรแกรมใช้งานลงในบอร์ด Arduino

เมื่ออัปโหลดเสร็จจะแสดงข้อความ Done uploading ดังรูปที่ 2.26



รูปที่ 2.26 ผลลัพธ์ เมื่ออัปโหลดลงบอร์ด Arduino สำเร็จ

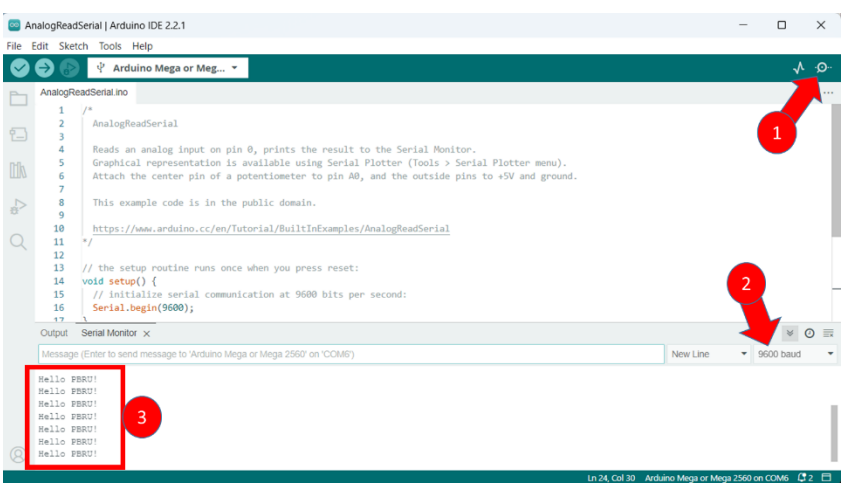
2.3.6 การใช้งาน Serial Monitor

Serial Monitor จะใช้ได้หลังจากเลือกพอร์ตแล้วเท่านั้น

ทดสอบเขียนโปรแกรมให้แสดงข้อความ Hello ทุก ๆ 1 วินาที ดังนี้

```
void setup() {
  // put your setup code here, to run once
Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly
  Serial.println("Hello PBRU !");
  delay(1000);
}
```

อัปโหลดโปรแกรมลงบอร์ดให้เรียบร้อย จากนั้นกดปุ่มรูปแว่นขยายที่มุมบนขวาของหน้าต่างโปรแกรม ปรับ baud rate เป็น 9600 จะนั้นจะมีข้อความ Hello แสดงทุก ๆ 1 วินาที ดังรูปที่ 2.27



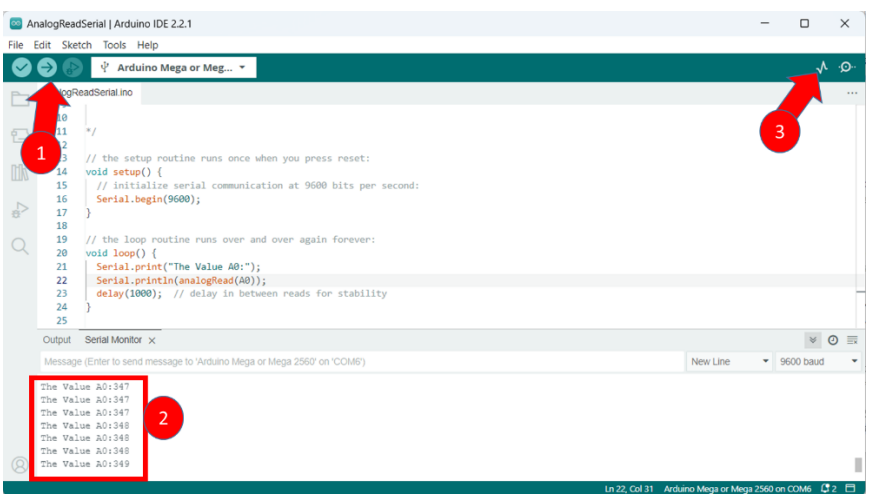
รูปที่ 2.27 ผลลัพธ์การเลือกใช้เมนู Serial Monitor

2.3.7 การใช้งาน Serial Plotter

Serial Plotter เป็นส่วนแสดงผลข้อมูลที่ส่งมาจากบอร์ดไมโครคอนโทรลเลอร์ในรูปแบบกราฟ จะใช้ได้หลังจากเลือกพอร์ตแล้วเท่านั้น ทดสอบเขียนโปรแกรมให้อ่านค่าอนาล็อกจากช่อง A0 ดังนี้

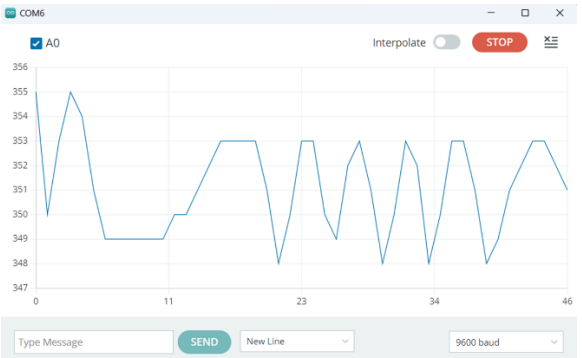
อัปโหลดโปรแกรม แล้วสังเกตใน Serial Monitor จะต้องมีค่าที่อ่านได้ขึ้น จากนั้นลองดูค่าในรูปแบบกราฟโดยกดรูปสัญญาณขึ้นลงที่มุมบนขวาของโปรแกรม ดังรูปที่ 2.28

```
void setup() {
  // put your setup code here, to run once
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly
  Serial.print("A0:");
  Serial.println(analogRead(A0));
  delay(100);
}
```



รูปที่ 2.28 เมนูในการเลือกใช้ Serial Plotter

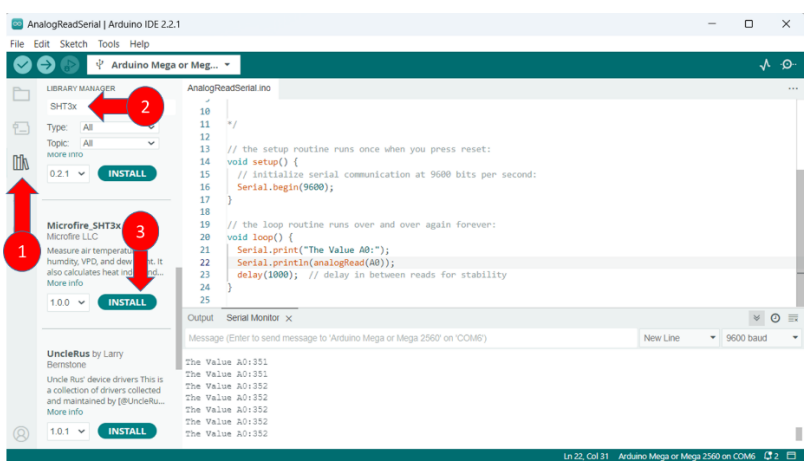
กราฟ (Serial Plotter) จะขึ้นมาดังรูปที่ 2.29



รูปที่ 2.29 ผลลัพธ์จาก Serial Plotter

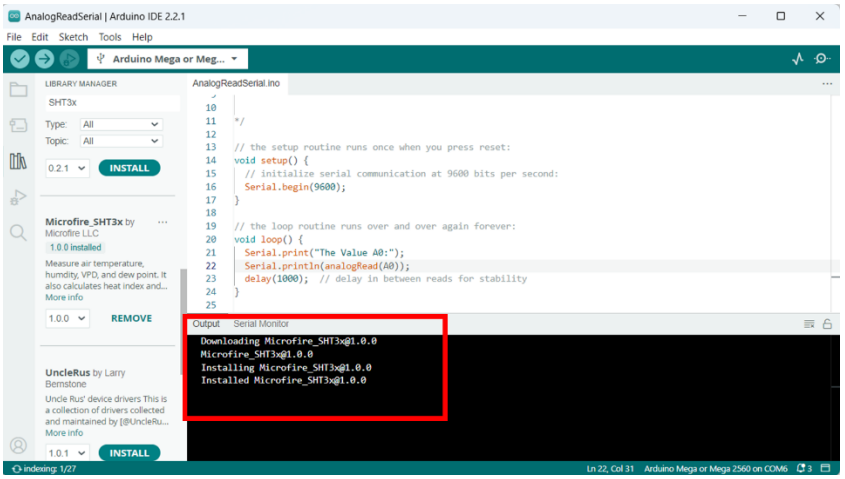
2.3.8 การติดตั้งไลบรารี

กดไอคอนรูปหนังสือที่มุมซ้าย แล้วพิมพ์ชื่อไลบรารีที่ต้องการติดตั้งในช่องค้นหา จากนั้นกดปุ่ม INSTALL ไลบรารีที่ต้องการ ดังรูปที่ 2.30



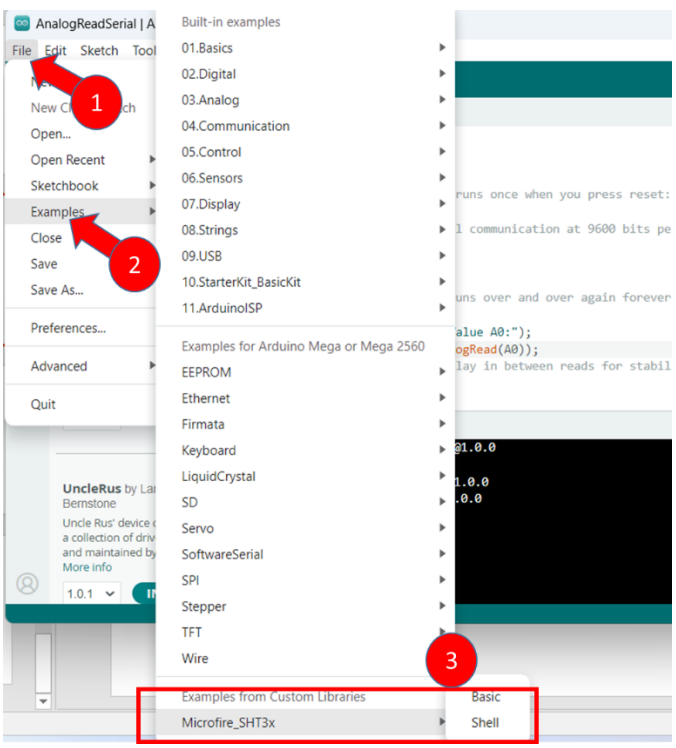
รูปที่ 2.30 วิธีการเลือกเมนูค้นหา Library และการติดตั้ง

เมื่อติดตั้งไลบรารีเสร็จ จะมีข้อความ Installed library ดังรูปที่ 2.31



รูปที่ 2.31 ผลลัพธ์ เมื่อติดตั้ง Library สำเร็จ

โปรแกรมตัวอย่างที่มีอยู่ในไลบรารี อยู่ในเมนู File > Examples เมนูย่อยชื่อไลบรารี ดังรูปที่ 2.32

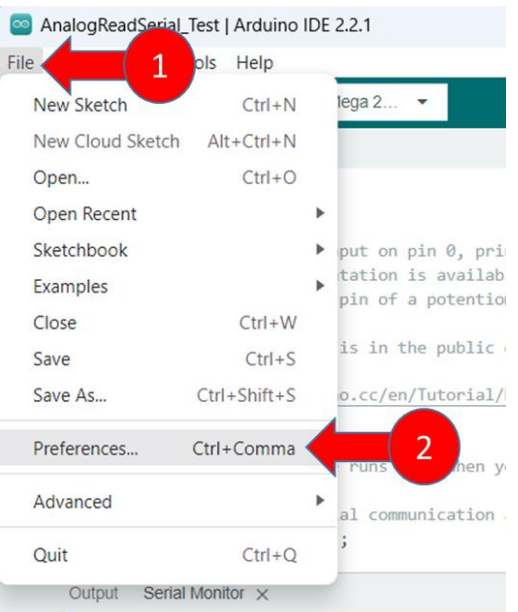


รูปที่ 2.32 วิธีการเลือกดูว่าไลบรารีที่ติดตั้งเรียบร้อยในโปรแกรม

2.3.9 การติดตั้งบอร์ด ESP32, ESP8266, STM32, RP2040 เพิ่ม

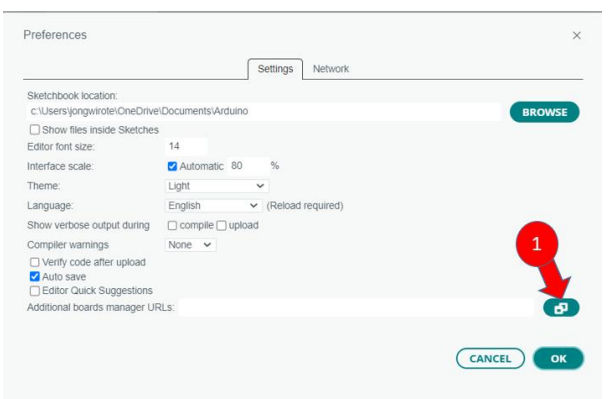
หากเคยติดตั้งโปรแกรม Arduino IDE V1 มาก่อน และเคยติดตั้งบอร์ดเพิ่มแล้ว บอร์ดที่เคยอยู่ในโปรแกรมเก่าจะมีให้เลือกในโปรแกรมโดยอัตโนมัติ หากยังไม่เคยติดตั้งบอร์ดมาก่อน ทำตามขั้นตอนดังรูปที่ 2.33 ดังนี้

กดเมนู File เลือก Preferences...



รูปที่ 2.33 วิธีในการเลือกเมนู Preferences

กดปุ่มด้านหลังหัวข้อ Additional boards manager URLs ดังรูปที่ 2.34

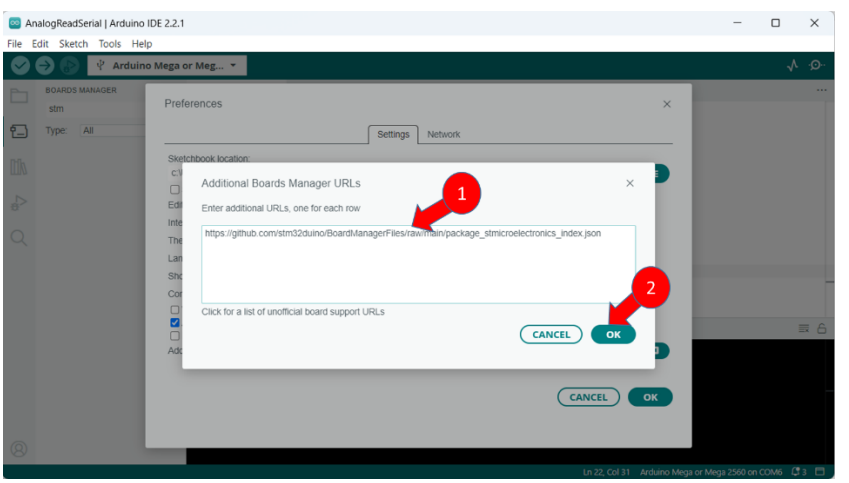


รูปที่ 2.34 วิธีการเลือก URL สำหรับการติดตั้งบอร์ดรุ่นอื่นๆ ที่ต้องการใช้งาน

ใส่เชื่อมโยงดาวน์โหลดบอร์ดที่ต้องการดังรูปที่ 2.35 โดยบอร์ดที่ต้องการติดตั้งเพิ่มจะมีเชื่อมโยงดาวน์โหลดไม่เหมือนกัน

- เพิ่มบอร์ด ESP32 ให้ใส่เชื่อมโยง https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json
- เพิ่มบอร์ด ESP8266, NodeMCU, WeMos ให้ใส่ เชื่อมโยง https://arduino.esp8266.com/stable/package_esp8266com_index.json
- เพิ่มบอร์ด STM32 ให้ใส่ เชื่อมโยง https://github.com/stm32duino/BoardManagerFiles/raw/main/package_stmicroelectronics_index.json
- เพิ่มบอร์ดใช้ชิป RP2040 ให้ใส่เชื่อมโยง https://github.com/earlephilhower/arduino-pico/releases/download/global/package_rp2040_index.json

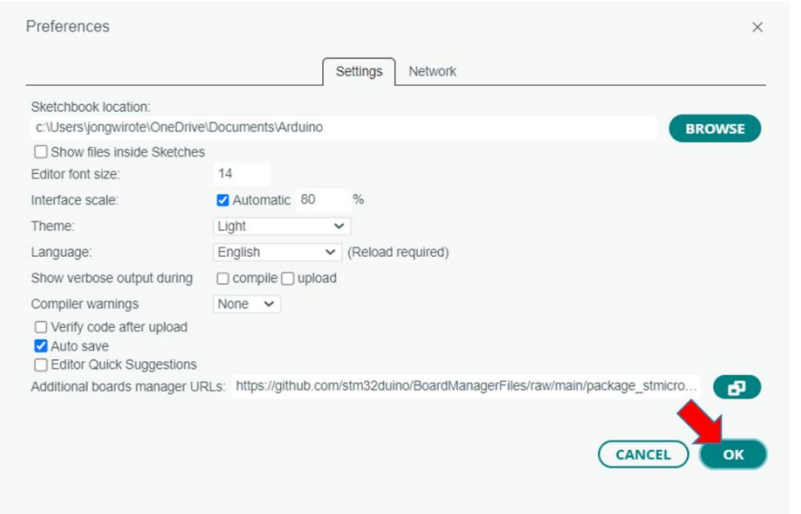
ตัวอย่างต้องการติดตั้งบอร์ด ESP32, ESP8266 และ STM32 เพิ่ม จึงใส่ 3 เชื่อมโยง (แต่ละเชื่อมโยงต้องแยกกันโดยกดขึ้นบรรทัดใหม่) แล้วกดปุ่ม OK



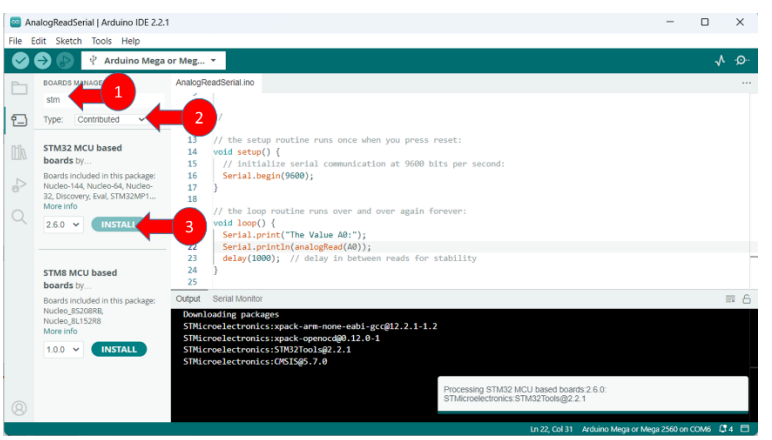
รูปที่ 2.35 วิธีการใส่ค่า URL เพื่อใช้สำหรับติดตั้งบอร์ดรุ่นอื่นๆ

กดปุ่ม OK อีกครั้งเพื่อบันทึกดังรูปที่ 2.36 แล้วหน้าต่าง Preferences จะหายไป

กดรูปบอร์ดที่เมนูด้านซ้าย (1) เลือก Type เป็น Contributed (2) จากนั้นเลื่อนหาบอร์ดที่ต้องการติดตั้ง (3) ดังรูปที่ 2.37 ตัวอย่างต้องการติดตั้งบอร์ด STM32 เพิ่ม จึงกดปุ่ม Install แล้วรอจนกว่าโปรแกรมจะติดตั้งบอร์ดเพิ่มเสร็จ (ขั้นตอนนี้อาจใช้เวลาหลายนาที ขึ้นอยู่กับความเร็วอินเตอร์เน็ต และความเร็วขอคอมพิวเตอร์เครื่องนั้น ๆ)



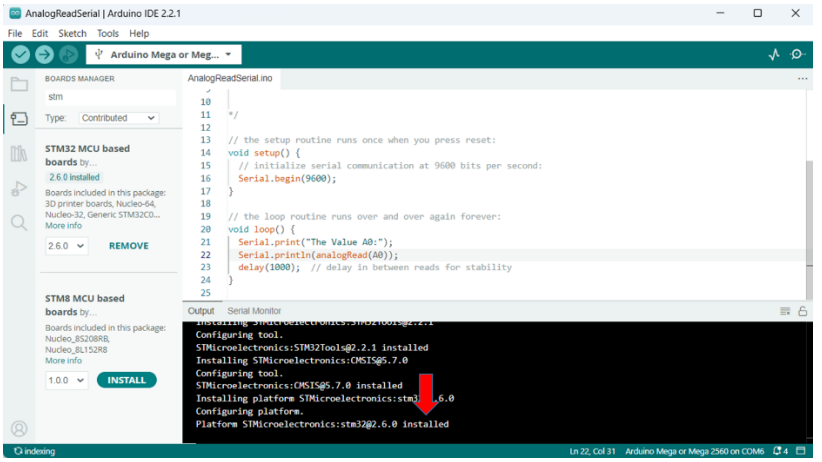
รูปที่ 2.36 เมนูเมื่อใส่ URL เรียบร้อย เพื่อดำเนินการต่อไป



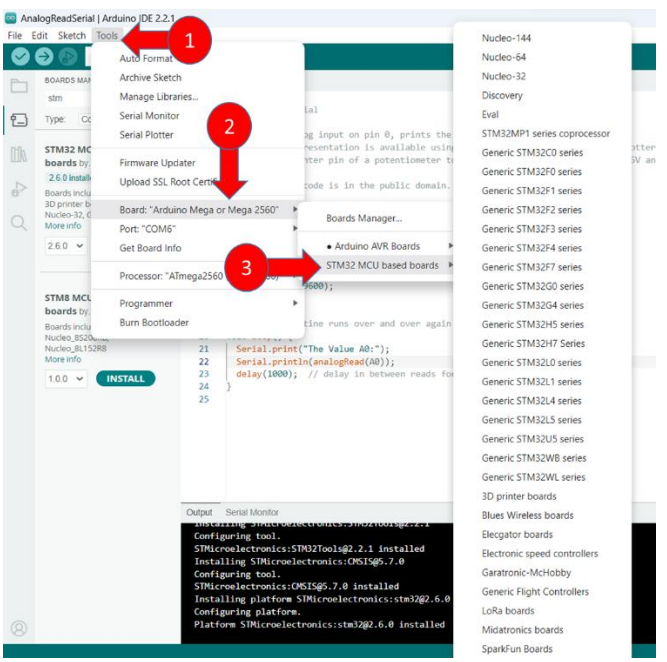
รูปที่ 2.37 วิธีเลือกในเมนู สำหรับการติดตั้งบอร์ดชนิดอื่นๆเพิ่มขึ้น

เมื่อติดตั้งเสร็จ จะแสดงข้อความ installed ขึ้นดังรูปที่ 2.38

ทดสอบเลือกบอร์ดโดยกด Tool > Board จะเห็นได้ว่ามีบอร์ดที่เพิ่งติดตั้ง (STM32) ดังรูปที่ 2.39 แสดงขึ้นมาให้เลือก



รูปที่ 2.38 ผลลัพธ์ ขณะเมื่อมีการติดตั้งบอร์ดที่เลือกใหม่



รูปที่ 2.39 วิธีการเลือกเมนู เพื่อแสดงให้บอร์ดที่ติดตั้งได้สำเร็จ

2.4 โปรแกรมภาษา C/C++ที่ใช้บน Arduino

โปรแกรมที่ใช้ในการเขียนควบคุมการทำงานของบอร์ด Arduino จะใช้โปรแกรมที่มีคำสั่งเสมือนกับการเขียนโปรแกรมภาษา C/C++ ซึ่งจะมีรูปแบบดังนี้

2.4.1 โครงสร้างของโปรแกรม

ในการเริ่มต้นเขียนโปรแกรมควบคุม Arduino IDE จะมีฟังก์ชันแสดงขึ้นมาสองส่วน ส่วนแรกจะเป็น void setup ( ) และส่วนที่สองจะเป็น void loop ( ) โดยที่จะมีการทำงานที่แตกต่างกัน ส่วนที่ void setup() จะทำงานเพียงครั้งเดียวเมื่อมีการใช้งานโปรแกรมในการเริ่มต้น ซึ่งจะใช้สำหรับการกำหนดค่าเริ่มต้น สำหรับการเขียนโปรแกรม ส่วนที่สอง void loop ( ) จะเป็นส่วนที่โปรแกรมจะทำงานอย่างต่อเนื่อง ซึ่งใช้เป็นส่วนที่จะควบคุมการทำงานบนบอร์ด Arduino ดังนี้

1. ฟังก์ชัน Setup ( )

ฟังก์ชัน Setup ( ) คือการประกาศค่าเริ่มต้นในการใช้โปรแกรม ตำแหน่งพอร์ตที่ใช้งาน ซึ่งจะรวมถึงไลบรารี่ที่ใช้งานด้วย จะทำงานเพียงครั้งเดียว และจะทำทุกครั้งที่มีการรีเซ็ต หรือรีบูตเครื่อง อาทิเช่น

```
 Void setup ( )
{ Serial.begin(9600);/* จะใช้งานพอร์ตอนุกรมที่ทำการสื่อสารกันด้วยความเร็ว 9600 บิต/วินาที*/
pinMode(Pin5, INPUT);//กำหนดใช้ขาดิจิตอลที่ 5สำหรับทำงานเป็นอินพุท
}
```

2. ฟังก์ชัน loop ( )

ฟังก์ชัน Loop ( ) คือ ส่วนที่โปรแกรมจะทำงานอย่างต่อเนื่องในการควบคุมบอร์ด Arduino ภายหลังจากที่มีการเรียกใช้โปรแกรม setup ( ) โดยที่ loop ( ) จะทำงานวนซ้ำ เพื่อให้โปรแกรมทำงานตามคำสั่งที่ต้องการ อาทิเช่น

```
Void loop ( )
{digitalWrite(buzzPin, HIGH);//คำสั่งที่ส่งค่าไฟเลี้ยงไปทำให้บัซเซอร์มีเสียงดังขึ้น
delay (100);
digitalWrite(buzzPin, LOW)://คำสั่งที่หยุดส่งไฟเลี้ยงไปที่บัซเซอร์ทำให้บัซเซอร์เงียบเสียง
delay(100);
}
```

2.4.2 การกำหนดตัวแปร

ในการเขียนโปรแกรมจำเป็นต้องมีการกำหนดตัวแปร เพื่อนำมาใช้ในการเขียนอ้างอิง และให้ค่ากับตัวแปรนั้น ซึ่งมีทั้งเป็นค่าคงที่ หรือมีการเปลี่ยนแปลงค่าตามที่ ผู้เขียนโปรแกรมต้องการกำหนด หรือให้มีการเปลี่ยนแปลงค่าตามที่ได้รับสัญญาณจากภายนอกได้ ซึ่งประกอบด้วยดังนี้

1.byte ตัวแปร byte ใช้เก็บตัวเลขขนาด 8 บิต ไม่มีทศนิยม มีค่าตั้งแต่ 0 ถึง 255

ตัวอย่างการเขียน byte Fvariable=99;// เป็นการกำหนดค่าตัวแปร Fvariable ให้เป็นตัวแปรชนิด byte

2.int ตัวแปร interger เป็นตัวแปรที่เก็บค่าจำนวนเต็ม และมีการเก็บค่าแบบ 16 บิต มีค่าตั้งแต่ -32,768 ถึง 32,768

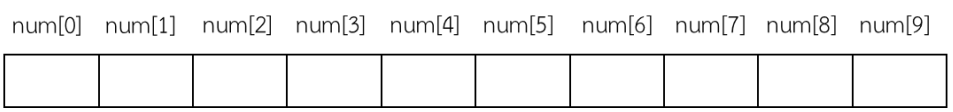
ตัวอย่างการเขียน int Fnumber=999;// เป็นกำหนดค่าตัวแปร Fnumber ให้เป็นตัวแปรชนิด int

3.long ตัวแปร long เป็นตัวแปรที่เก็บค่าจำนวนเต็มที่มีค่ากำหนดได้มากกว่า int เก็บค่าแบบ 32 บิต มีค่าตั้งแต่ -2,147,483,648 ถึง 2,147,483,648

4.float ตัวแปร float เป็นตัวแปรที่เก็บค่าจำนวนที่มีจุดทศนิยม เก็บค่าแบบ 32 บิต มีค่าตั้งแต่ -3.4028235E+38 ถึง 3.4028235E+38

5.array ตัวแปร array เป็นตัวแปรชนิดแถวลำดับ ตัวแปรชนิดแถวลำดับแบบ 1มิติ จะเป็นการเก็บข้อมูลแบบแถวต่อเนื่องกัน ยกตัวอย่าง

int num[10] จะมีความหมายว่าเตรียมไว้ 10 ช่องสำหรับการใช้เลขจำนวนเต็ม ดังรูปที่ 2.40



รูปที่ 2.40 การเตรียม 10 ช่องสำหรับใส่ข้อมูล

2.4.3 ตัวดำเนินการทางคณิตศาสตร์

ตัวดำเนินการหรือตัวดำเนินการทางคณิตศาสตร์นั้น มีหลายตัว นักเขียนโปรแกรม จำเป็นต้องทราบว่า ในโปรแกรมนั้น จะมีการรับรู้ว่าเครื่องหมายแต่ละตัวมีความหมายอย่างไร โดยผู้เขียนจะต้องทราบความหมาย และนำมาใช้งาน ซึ่งตัวดำเนินการหลักๆ จะมีอยู่ 5 ตัวด้วยกัน แสดงได้ดังตารางที่ 2.1

ตารางที่ 2.1 ตัวดำเนินการทางคณิตศาสตร์

ตัวดำเนินการ	ความหมาย
+	การบวก(addition)
-	การลบ(subtraction)
*	การคูณ(multiplication)
/	การหาร(division)
%	การหารเอาเศษ(modulus)

ตัวอย่างที่ 2.1 กำหนดให้ x และ y เป็นตัวแปรชนิดเลขจำนวนเต็มทั้งสองตัว ซึ่งผู้เขียนโปรแกรม จะกำหนดค่าเป็น 21 และ 2 ตามลำดับ เมื่อนำตัวแปรทั้งสองมาผ่านนิพจน์คณิตศาสตร์จะได้ผลลัพธ์ตามตารางที่ 2.2

ตารางที่ 2.2 ตัวดำเนินการผ่านนิพจน์คณิตศาสตร์

นิพจน์	ผลลัพธ์
x + y	23
x –y	19
x * y	42
x / y	10
x % y	1

2.4.4 ตัวดำเนินการความสัมพันธ์

ตัวดำเนินการความสัมพันธ์จะแสดงการเปรียบเทียบนิพจน์ 2 นิพจน์ ที่ประสงค์ให้มีการเปรียบเทียบด้วยเครื่องหมายดังตารางที่ 2.3 และแสดงความหมายในเชิงว่า มากกว่า น้อยกว่า มากกว่าหรือเท่ากับ เป็นต้น

ตารางที่ 2.3 ตัวดำเนินการที่ใช้เปรียบเทียบนิพจน์

ตัวดำเนินการ	ความหมาย
>	มากกว่า
<	น้อยกว่า
>=	มากกว่าหรือเท่ากับ
<=	น้อยกว่าหรือเท่ากับ

ซึ่งผลที่ได้จากการเปรียบเทียบความสัมพันธ์ จะแสดงผลเป็นตรรกะว่า เป็นความจริง หรือความเท็จ ซึ่งจะแสดงเป็นเลขจำนวนเต็ม ค่า 0 หมายถึงว่าผลการเปรียบเทียบนิพจน์เป็นเท็จ แต่ถ้าค่าเป็น 1 หมายถึงว่าผลการเปรียบเทียบค่านิพจน์เป็นจริง สามารถแสดงผลของความสัมพันธ์ของนิพจน์ได้ดังตารางที่ 2.4

ตารางที่ 2.4 การประมวลผลนิพจน์ความสัมพันธ์ และผลลัพธ์ที่ได้

นิพจน์	ใส่ตัวเลขในนิพจน์	ผลลัพธ์ตรรกะ	ผลลัพธ์ที่ได้
a+b*c<(a+b)*c	1+(2*3)<(1+2)*3	จริง	1
x+y <= y+x	(8+4)<=(4+8)	จริง	1
x/y <y/x	(4/2)<(2/4)	เท็จ	0

2.4.5 ตัวดำเนินการความเท่ากัน

ตัวดำเนินการความเท่ากัน จะเป็นเปรียบเทียบนิพจน์ว่ามีความเท่ากัน หรือไม่เท่ากัน ซึ่งจะแสดงความหมายได้ดังตารางที่ 2.5

ตารางที่ 2.5 ตัวดำเนินการเพื่อใช้เปรียบเทียบความเท่ากันของนิพจน์

ตัวดำเนินการ	ความหมาย
= =	เท่ากัน
! =	ไม่เท่ากัน

โดยผลลัพธ์ของการเปรียบเทียบจะแสดงผลลัพธ์เป็นค่าจริง หรือเท็จ

2.4.6 ตัวดำเนินการตรรกะ

ตัวดำเนินการตรรกะ จะเป็นการดำเนินความสัมพันธ์ที่จะมีความหมายได้ดังแสดงในตารางที่ 2.6 ดังต่อไปนี้

ตารางที่ 2.6 ตัวดำเนินการตรรกะ

ตัวดำเนินการตรรกะ	ความหมาย
&&	และ (and)
||	หรือ (or)
!	ไม่ใช่ (not)

ซึ่งผลลัพธ์ที่ได้จากการเปรียบเทียบความสัมพันธ์ตรรกะ จะแสดงผลลัพธ์เป็นตรรกะว่าเป็นจริง หรือเป็นเท็จ ซึ่งจะแสดงค่า 0 ว่าผลการเปรียบเทียบเป็นเท็จ หากแสดงค่า 1 จะถือว่าผลการเปรียบเทียบเป็นจริง สามารถดูรายละเอียดได้ดังตารางที่ 2.7 ดังนี้

ตารางที่ 2.7 การเปรียบเทียบตรรกะ และการแสดงค่าความจริง

ตัวแปรที่ใช้ในการดำเนินการเปรียบเทียบตรรกะ		การแสดงค่าความจริงของผลลัพธ์ที่ได้		
X	Y	X&&Y	X||Y	! X
1	1	1	1	0
1	0	0	1	0
0	1	0	1	1
0	0	0	0	1

2.4.7 คำสั่งในการทำซ้ำ

คำสั่ง for loop เป็นคำสั่งที่ใช้งานง่าย มีการกำหนดรอบทำซ้ำว่าจะต้องทำกี่รอบ รูปแบบของ for loop จะมีลักษณะการเขียนดังนี้

```
for (ค่าเริ่มต้น ; เงื่อนไขตรวจสอบ ; เพิ่มหรือลดจำนวน )
{
// ชุดคำสั่งกรณีที่เงื่อนไขตรวจสอบเป็นจริง
คำสั่ง..... 1
คำสั่ง..... 2
คำสั่ง..... 3
.
.
.
คำสั่ง..... n
```

```
}
```

ตัวอย่างที่ 2.2 เป็นการสั่งกำหนดค่าเริ่มต้นที่ตัวแปร i มีค่าเป็นจำนวนเต็มเท่ากับ หนึ่ง ทดสอบค่า i ถ้ามีค่าน้อยกว่าหรือเท่ากับ 10 จะให้ทำคำสั่งในวงเล็บปีกกา แล้วบวกหนึ่งเพิ่มในค่า i ทดสอบจนกระทั่งค่า i มีค่ามากกว่า 10 จึงจะออกจากคำสั่งวนลูป

```
for (int i=1; i<=10; i++)

{

DigitalWrite(12,HIGH);

}
```

<u>คำสั่ง while loop</u> เป็นคำสั่งที่ใช้ในกรณีที่ไม่ทราบจำนวนการทำซ้ำที่แน่นอน แต่ลักษณะเด่นของคำสั่ง while คือ มีการตรวจสอบเงื่อนไขว่าเป็นจริงก่อนการทำคำสั่ง หากเงื่อนไขตรวจสอบเป็นเท็จ ก็จะไม่มีการทำซ้ำเลย แต่ในทำนองเดียวกันถ้าตัวตรวจสอบเงื่อนไขเป็นจริงตลอด โปรแกรมจะทำซ้ำไม่สิ้นสุด

```
while (เงื่อนไขตรวจสอบ)

{

// ชุดคำสั่งกรณีที่เงื่อนไขตรวจสอบเป็นจริง

คำสั่ง..... 1

คำสั่ง..... 2

คำสั่ง..... 3

.
.
.

คำสั่ง..... n

}
```

ตัวอย่างที่ 2.3 แสดงการใช้คำสั่ง while loop

```
while ( i <=20)

{
```

```
digitalWrite(9,HIGH);

i++;

}
```

<u>คำสั่ง do while</u> เป็นคำสั่งการทำซ้ำที่มีลักษณะการทำงานเช่นเดียวกับคำสั่ง while แตกต่างกันที่คำสั่ง do while จะมีการทำงานตามคำสั่งไปก่อน 1 รอบ แล้วจึงทดสอบเงื่อนไข หากเงื่อนไขเป็นจริงจะทำงานต่อไป แต่หากเป็นเท็จจะออกจากคำสั่งทันที

รูปแบบคำสั่ง do while จะมีลักษณะการเขียนดังนี้

```
do

{

คำสั่ง..... 1

คำสั่ง..... 2

คำสั่ง..... 3

.
.
.

คำสั่ง..... n

} while (เงื่อนไขตรวจสอบ);
```

ตัวอย่างที่ 2.4 แสดงการใช้คำสั่ง do while

```
do

{

m = readSensor( );

delay(1000);

}
```

while(m <100);

2.4.8 คำสั่งจำเพาะ

Constants

เป็นคำสั่งที่ใช้กำหนดค่าคงที่ให้กับตัวแปรที่อยู่ในโปรแกรม

True/False

เป็นการกำหนดค่าความจริงให้ข้อมูล เลข 0 หมายถึง False และเลข 1 หมายถึง True

High/Low

เป็นการกำหนดค่าระดับแทนแรงดันไฟฟ้า ถ้าจะให้มีค่า HIGH จะแทนระดับแรงดัน 5 โวลท์ และค่า LOW จะแทนระดับแรงดัน 0 โวลท์

Input/Output

เป็นการกำหนดค่าในฟังก์ชัน pinMode() เพื่อกำหนดหน้าที่ของขา ที่จะทำหน้าที่เป็นอินพุท หรือเอาท์พุท

pinMode(pin,mode)

ยกตัวอย่างเช่น pinMode(12,INPUT); หมายถึงว่ากำหนดให้ขา 12 มีหน้าที่เป็นอินพุท

digitalRead(pin)

เป็นคำสั่งที่ใช้อ่านค่าจากขา ที่ทำหน้าที่เป็นขาในการรับส่งข้อมูลแบบดิจิทัล โดยข้อมูลที่ได้จะเป็น HIGH หรือ LOW โดยมีรูปแบบดังนี้

value=digitalRead(8);// กำหนดให้ตัวแปร value มีค่าเท่ากับข้อมูลที่ได้รับจากขา 8

digitalWrite(pin)

เป็นคำสั่งที่ใช้ในการส่งค่าให้ออกจากขา เพื่อส่งข้อมูลออก มีรูปแบบดังนี้

digitalWrite(8,HIGH); // กำหนดขา 8 ให้มีค่าเท่ากับ HIGH

analogRead(pin)

เป็นคำสั่ง เพื่อใช้อ่านค่าจากขาอนาล๊อค โดยที่บอร์ด Arduino จะมีวงจร A/D ขนาด 10 บิต เพื่อแปลงค่าแรงดันที่ผ่านได้จากขาอนาล๊อค โดยผลลัพธ์จะเป็นเลขจำนวนเต็มมีค่าระหว่าง 0-1023 โดยรูปแบบคำสั่งดังนี้

Value = analogRead(pin);

analogWrite(pin, value)

เป็นคำสั่งที่ให้ค่าแรงดันที่ขาอนาล๊อค เปลี่ยนไปตามข้อมูลที่กำหนด เพื่อให้ได้สัญญาณในรูปของ Pulse Width Modulation(PWM) ถ้าค่า 0 แรงดันที่ขาอนาล๊อค จะเป็น 0 โวลท์ ส่วนค่า 255 จะเป็น 5 โวลท์ แต่ค่าระหว่าง 0-255 ค่าแรงดันที่ขาอนาล๊อคจะเปลี่ยนไปเป็นค่าที่อยู่ระหว่าง 0 ถึง 5 โวลท์ โดยที่ค่ายิ่งมากแรงดันที่ขาอนาล๊อคจะมีค่าเข้าใกล้5 โวลท์

delay(ms)

เป็นคำสั่งที่ใช้ในการตั้งค่าหน่วงเวลาการทำงานให้กับโปรแกรม โดยกำหนดค่าเป็นหน่วยมิลลิวินาที โดยที่ 1000 จะมีค่าเท่ากับ 1 วินาที

millis()

เป็นคำสั่งที่กำหนดให้ผลลัพธ์ได้ค่าเวลาเป็นมิลลิวินาที โดยมีรูปแบบดังนี้

Value =millis (100);// กำหนดให้ค่าตัวแปร value มีค่าเท่ากับ 100 มิลลิวินาที

Serial.begin (rate)

เป็นคำสั่งที่ใช้เปิดพอร์ตอนุกรม และกำหนดค่าความเร็วในการรับส่งข้อมูล โดยกำหนดความเร็วได้ มีรูปแบบดังนี้

void setup

{

Serial.begin(9600);}// เปิดพอร์ตอนุกรม และกำหนดค่าความเร็วในการรับส่งข้อมูลมีค่าเท่ากับ 9,600 บิตต่อวินาที

Serial.print(data)

เป็นคำสั่งที่ส่งข้อมูลให้กับพอร์ตอนุกรม และสามารถที่จะเรียกดูข้อมูลได้ที่ เมนู Serial monitor โดยมีรูปแบบดังนี้

Serial.print(analogvalue);// ส่งค่าของตัวแปร analogvalue ไปที่พอร์ตอนุกรม

Serial.println(data)

เป็นคำสั่งที่ส่งข้อมูลให้กับพอร์ตอนุกรม แล้วตามด้วยการขึ้นบรรทัดใหม่โดยอัตโนมัติ คำสั่งนี้จะมีผลเหมือนกับคำสั่ง Serial.print ( ) และสามารถเรียกดูข้อมูลได้ในเมนู Serial Monitor โดยมีรูปแบบดังนี้

Serial.println(analogvalue); //ส่งค่าของตัวแปร analogvalue ไปที่พอร์ตอนุกรม

สรุปท้ายบท

สรุปได้ว่า Arduino IDE V2 และ V1 มีหน้าตาโปรแกรม (UI) ที่แตกต่างกันพอสมควร แต่ทางผู้พัฒนาก็พยายามจะใช้คำศัพท์ใน V2 ให้เหมือน V1 เพื่อให้ผู้ใช้สามารถเชื่อมโยงและเข้าใจได้ โปรแกรมควบคุมบอร์ด Arduino เป็นโปรแกรมที่ใช้ในการพัฒนา สรุปได้ดังนี้

1. Arduino IDE โปรแกรมควบคุมบอร์ด Arduino เป็น IDE ที่เข้าใจง่ายและใช้งานสะดวกสำหรับนักพัฒนา
2. เขียนโปรแกรม ใช้สำหรับเขียนโปรแกรมในภาษา Arduino ที่เรียกว่า "สคริปต์" ซึ่งเป็นภาษาโปรแกรมที่เรียบง่ายและเข้าใจ
3. อัปโหลดโปรแกรม ใช้สำหรับอัปโหลดโปรแกรมที่เขียนลงบอร์ด Arduino เพื่อให้บอร์ดสามารถทำงานตามโปรแกรมที่เขียนขึ้น
4. โหลดไลบรารี สามารถเพิ่มความสามารถในการพัฒนาโปรแกรมด้วยการโหลดไลบรารีที่เหมาะสำหรับโพรเจกต์ของผู้อ่าน
5. ติดตามข้อผิดพลาด IDE มีเครื่องมือในการตรวจสอบข้อผิดพลาดในโปรแกรมและช่วยให้ผู้อ่านแก้ไขโปรแกรมได้อย่างง่าย
6. มีชุมชนและเอกสารมากมายที่สนับสนุนและช่วยในการพัฒนา และแก้ไขปัญหาที่เกิดขึ้น
7. เหมาะสำหรับการเริ่มต้นการเรียนรู้ ลงทุนที่น้อย โปรแกรมนี้เหมาะสำหรับนักพัฒนาทั้งมืออาชีพและผู้ที่เริ่มต้น

โปรแกรมควบคุมบอร์ด Arduino เป็นเครื่องมือสำคัญสำหรับนักพัฒนาในการพัฒนาและทดสอบโปรแกรมบอร์ด Arduino และอุปกรณ์อิเล็กทรอนิกส์ต่าง ๆ ที่ใช้ในโพรเจกต์ต่าง ๆ

คำถามท้ายบท

1. โปรแกรมที่ใช้ในการเขียนควบคุมบอร์ด Arduino มีกี่แบบ อะไรบ้าง
2. ฟังก์ชัน `setup()` ในโปรแกรม Arduino IDE ใช้ทำอะไร
3. ฟังก์ชัน `loop()` ในโปรแกรม Arduino IDE ใช้ทำอะไร
4. คำสั่ง `pinMode(pin, mode)` ใช้ทำอะไร
5. คำสั่ง `digitalWrite(pin, value)` ใช้ทำอะไร

เอกสารอ้างอิง

1. เดชฤทธิ์ มณีธรรม, *คัมภีร์การใช้งานไมโครคอนโทรลเลอร์ Arduino*, กรุงเทพฯ: ซีเอ็ดยูเคชั่น, 2560.

2. สุวิทย์ เมาะราษี, *การโปรแกรมมิ่งบอร์ด Arduino ด้วย C# .NET และ Sketch*, สำนักพัฒนาสมรรถนะครูและบุคลากรอาชีวศึกษา, 2563.

3. "*Arduino - Home*," Arduino.cc. [Online]. Available: https://www.arduino.cc/., 2023.

4. ศิริพงษ์ ฉายสินธ์, *การใช้งาน Arduino เบื้องต้น*, กรุงเทพฯ: มหาวิทยาลัยศรีนครินทรวิโรฒ, 2562.

5. S. Fitzgerald and M. Shiloh, *The Arduino Projects Book*, Torino, Italy: Arduino, 2012.

6. J. Blum, *Exploring Arduino: Tools and Techniques for Engineering Wizardry*, Hoboken, NJ: Wiley, 2013.

7. ทศพล บ้านคลองสี่, *Practical Microcontroller Programming with Arduino*, กรุงเทพฯ: ไอดีซีพรีเมียร์, 2565.

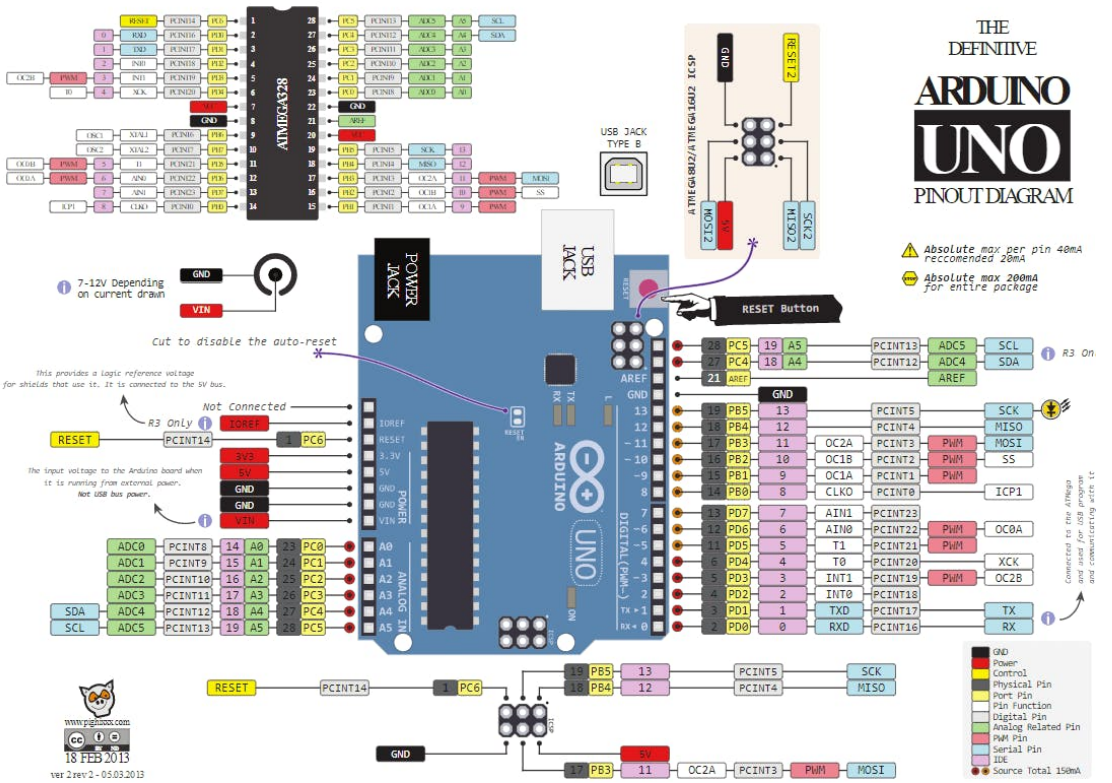
8. https://www.arduino.cc/en/software: Accessed:Oct.23, 2023.

บทที่ 3 ภาคทฤษฎีในการควบคุมขาอินพุตและเอาต์พุตของ Arduino

ในการทำงานควบคุมฮาร์ดแวร์ผ่าน Arduino จำเป็นต้องทราบว่าขาอินพุตและขาเอาต์พุต ที่จะทำการออกแบบให้เป็นส่วนในการรับสัญญาณเข้า และการส่งสัญญาณออก เมื่อไมโครโปรเซสเซอร์ได้ทำการประมวลผลที่ต้องการเรียบร้อยแล้ว ซึ่งผู้อ่านจะทราบรายละเอียดการทำงานได้ ดังต่อไปนี้

3.1การควบคุมขาอินพุต และขาเอาต์พุต

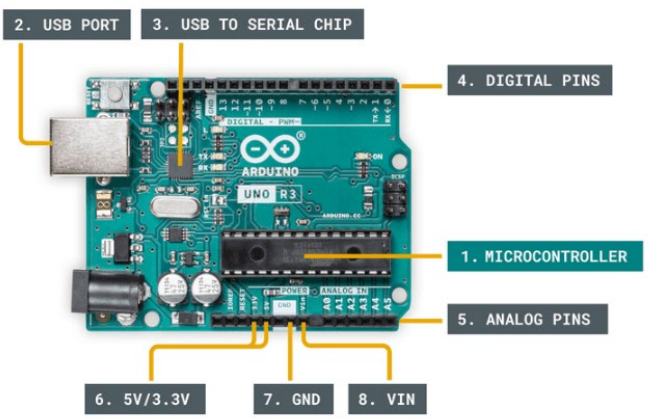
การควบคุมขาอินพุต และขาเอาต์พุต ของ Arduino เป็นสิ่งสำคัญในการสร้างโครงการอิเล็กทรอนิกส์และการโปรแกรมโดยใช้บอร์ด Arduino ขาอินพุตและขาเอาต์พุตเป็นขาที่สามารถใช้ในการรับข้อมูลจากเซ็นเซอร์หรือสัญญาณอินพุตและส่งข้อมูลไปยังอุปกรณ์หรือควบคุมอุปกรณ์อื่น ๆ ที่ต่อกับบอร์ด Arduino ดังรูปที่ 3.1



http://www.engineer007.com/index.php?lite-article&qid=42096260: Accessed: Jul.2, 2024

รูปที่ 3.1สถาปัตยกรรมของบอร์ด Arduino

จะแสดงว่าบอร์ด Arduino เป็นแพลตฟอร์มฮาร์ดแวร์ที่พัฒนาขึ้นเพื่อใช้ในการสร้างโพรเจกต์ที่เกี่ยวกับการควบคุมและประมวลผลข้อมูลต่าง ๆ โดยใช้ไมโครคอนโทรลเลอร์ จากรูปที่ 3.2 จะพบว่ามีพอร์ตอินพุต/เอาต์พุต สำหรับการเชื่อมต่อกับอุปกรณ์ภายนอก ในส่วนของอินพุต และ เอาต์พุตบนบอร์ด Arduino มีลักษณะดังนี้



รูปที่ 3.2 ขาที่ใช้ในบอร์ด Arduino UNO

1. อินพุต

- ดิจิทัลอินพุต บอร์ด Arduino มีขาดิจิตอลอินพุตที่ใช้สำหรับการรับสัญญาณดิจิตอล (ON/OFF) จากอุปกรณ์ภายนอก เช่น สวิตช์ หรือเซ็นเซอร์ แบบดิจิตอล

- อนาล็อกอินพุต บอร์ด Arduino มีขาอนาล็อกอินพุตที่ใช้สำหรับการรับสัญญาณอนาล็อก ซึ่งสามารถรับค่าอนาล็อกในช่วงตัวเลข 0-1023 (หรือ 0-255 ในบางรุ่น) จากเซ็นเซอร์แบบอนาล็อก เช่น เซ็นเซอร์แสดงอุณหภูมิหรือแสง

- การสื่อสารแบบอนุกรม(Serial Communication) Arduino สามารถเชื่อมต่อกับอุปกรณ์อื่นผ่านพอร์ตอนุกรม อาทิเช่น USB, UART, I2C เป็นต้น เพื่อรับข้อมูลจากเซ็นเซอร์หรืออุปกรณ์อื่น ๆ

2. เอาต์พุต

- ดิจิทัลเอาต์พุต บอร์ด Arduino มีขาดิจิตอลเอาต์พุตที่ใช้สำหรับควบคุมอุปกรณ์ดิจิตอล เช่น LED, มอเตอร์, รีเลย์, หรืออุปกรณ์อื่น ๆ ที่มีสถานะเปิด/ปิด (ON/OFF)

- PWM Outputs บอร์ด Arduino มีขาเอาต์พุต PWM ที่ใช้สำหรับควบคุมความสว่างหรือความเร็วของอุปกรณ์ เช่น การควบคุมความสว่างของ LED หรือควบคุมความเร็วของมอเตอร์

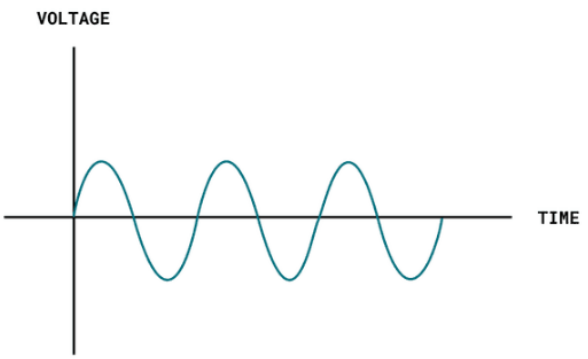
- อนาล็อกเอาต์พุต บอร์ด Arduino สามารถใช้ขาอนาล็อกเอาต์พุตเพื่อสร้างสัญญาณอนาล็อกเช่นควบคุมความเร็วของมอเตอร์หรือเสียงของลำโพง

การเชื่อมต่อและใช้งานขาอินพุตและเอาต์พุตบนบอร์ด Arduino จะขึ้นอยู่กับโครงสร้างของโพรเจกต์และประเภทของอุปกรณ์ที่ผู้อ่านต้องการควบคุมหรือส่งข้อมูลเข้าหรือออก Arduino มีการเขียนโปรแกรมแบบ Arduino IDE ที่ใช้ในการควบคุมขาอินพุตและเอาต์พุตของบอร์ด ลักษณะคล้ายภาษาโปรแกรม C/C++

3.2 สัญญาณที่ใช้สำหรับการควบคุมขาอินพุตและเอาต์พุต

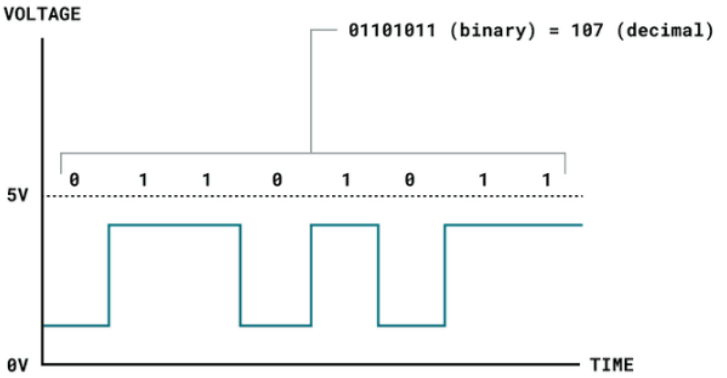
สัญญาณที่ใช้จะมี 2 ลักษณะสัญญาณ คือ สัญญาณอนาล็อก และสัญญาณดิจิทัล

จากรูปที่ 3.3 สัญญาณอนาล็อกทั่วไปมักจะมีขอบเขตของค่าในการทำงาน ใน Arduino ระหว่าง 0-5V หรือ 0-3.3V เป็นที่นิยม ถ้าใช้โพเทนชิโอมิเตอร์ (อุปกรณ์อนาล็อกที่ใช้ในการเปลี่ยนค่าความต้านทานในวงจร), สามารถปรับเปลี่ยนช่วงค่านี้ (0-5V) ด้วยผู้ใช้งาน ในโปรแกรมจะใช้ด้วยช่วงค่า 0-1023 ซึ่งเป็นความละเอียด 10 บิต ถ้าเขียนสัญญาณอนาล็อกโดยใช้เทคนิค Pulse-Width Modulation (PWM) สามารถใช้ช่วงค่าระหว่าง 0-255, เนื่องจากจะใช้ความละเอียด 8 บิต



รูปที่ 3.3 ลักษณะสัญญาณอนาล็อก

จากรูปที่ 3.4 สัญญาณดิจิทัลทำงานเล็กน้อยแตกต่างกันโดยที่มีเพียงสองสถานะทวิภาค (0 หรือ 1) ซึ่งอ่านให้เป็นสถานะสูงหรือต่ำในโปรแกรม ผู้ใช้งานสามารถอ่านและเขียนสัญญาณดิจิทัลอย่างง่ายในบอร์ด Arduino ซึ่งมีประโยชน์ในการอ่านสถานะปุ่ม เช่น เปิดหรือปิดอุปกรณ์ที่ต้องการ สัญญาณดิจิทัล เหมือนพื้นฐานของสัญญาณ ที่แสดงค่า 0 หรือ 1 แต่ในความเป็นจริง มีความสามารถที่ซับซ้อนยิ่งขึ้น ตัวอย่างเช่น สามารถสร้างสัญญาณลำดับโดยการส่งสถานะสูงหรือต่ำอย่างรวดเร็วหลายครั้งต่อเนื่องกัน หรือเรียกว่าลำดับทวิภาคหรือบิตสตรีม (bit stream) ได้



รูปที่ 3.4 ลักษณะสัญญาณดิจิทัล

3.3 คำสั่งในการรับขาอินพุต และคำสั่งในการส่งออกขาเอาต์พุต

บอร์ด Arduino มีหลายขาอินพุตและเอาต์พุตที่สามารถใช้ในการอ่านข้อมูลเข้าและเขียนข้อมูลออก โดยทั่วไปแล้วแบ่งเป็นสองประเภทคืออนาล็อกและดิจิทัล

1. อนาล็อกอินพุต

- ขาอินพุตอนาล็อกใน Arduino เช่น A0, A1, A2 เป็นต้น ทำหน้าที่อ่านค่าอนาล็อก เช่น ค่าสถานะของตัวแปรโดยใช้เซ็นเซอร์อนาล็อกหรือความต้านทานที่เปลี่ยนไป
- คำสั่งอนาล็อกอินพุต

```
int value = analogRead(A0); // อ่านค่าอนาล็อกจากขา A0
```

2. อนาล็อกเอาต์พุต

- บอร์ด Arduino มีขาเอาต์พุตอนาล็อก ซึ่งสามารถใช้ในการส่งสัญญาณอนาล็อกเช่น PWM (Pulse-Width Modulation)
- คำสั่งอนาล็อกเอาต์พุต

```
analogWrite(9, 128); // ส่งสัญญาณ PWM ที่ระดับความสูง 128 ไปยังขาดิจิทัล 9
```

3. ดิจิทัลอินพุต

- ขาอินพุตดิจิทัลใน Arduino เช่น 1,2, 3, 4, 5 เป็นต้น ใช้ในการอ่านสถานะดิจิทัลเช่น 0 (LOW) หรือ 1 (HIGH) จากปุ่มสวิทช์หรือเซ็นเซอร์ดิจิทัล
- คำสั่งอินพุตดิจิทัล

```
int buttonState = digitalRead(2); // อ่านสถานะจากขาดิจิทัล 2
```

4. ดิจิทัลเอาต์พุต
- ขาดิจิทัลเอาต์พุต เช่น 6, 7, 8 เป็นต้น ใช้ในการส่งสัญญาณดิจิทัลเช่น 0 (LOW) หรือ 1 (HIGH) เพื่อควบคุมอุปกรณ์อิเล็กทรอนิกส์เช่น LED หรือมอเตอร์
- คำสั่งดิจิทัลเอาต์พุต

```
digitalWrite(6, HIGH); // ส่งสัญญาณ HIGH ไปยังขาดิจิทัล 6
```

สิ่งสำคัญคือต้องเชื่อมต่ออุปกรณ์หรือเซ็นเซอร์กับขาที่ถูกตั้งค่าให้ถูกต้องในโปรแกรม เพื่อรับค่าและส่งค่าสัญญาณ หรือติดต่อสื่อสารกันได้ก่อน และบอร์ด Arduino สามารถที่ใช้ค่าสัญญาณอนาล็อกหรือสัญญาณดิจิทัล จากอุปกรณ์เซ็นเซอร์หรือสถานะของอุปกรณ์ เพื่อควบคุมและจำแนกสถานะ และสามารถเชื่อมต่อกับโลกดิจิทัล หรืออินเทอร์เน็ตของสรรพสิ่งได้

3.4 วิธีการควบคุมขาอินพุตและขาเอาต์พุตของ Arduino

จากหัวข้อข้างต้น จะสามารถเข้าใจคำสั่งในการควบคุมขาอินพุตและขาเอาต์พุตได้ เพื่อให้เกิดความกระจ่างชัดเจน จึงขอนำเสนอแนวทางการเขียนโปรแกรม ที่สามารถใช้งานในการเขียนโปรแกรมควบคุมขาทั้งสอง ได้ดังนี้

1. ขาอินพุตของ Arduino ใช้ในการรับข้อมูลจากเซ็นเซอร์หรืออุปกรณ์อื่นที่ส่งสัญญาณเข้ามาในบอร์ด จะใช้ฟังก์ชัน `digitalRead()` เพื่ออ่านค่าของขาอินพุต เป็นค่าดิจิทัล (Digital) ซึ่งอาจเป็น 0 (LOW) หรือ 1 (HIGH)

ตัวอย่างในการอ่านขาอินพุต

```
int sensorPin = 2;  // กำหนดขาอินพุตที่เซ็นเซอร์ต่อเข้า

int sensorValue = 0; // ตัวแปรสำหรับเก็บค่าอินพุต

void setup() {

  pinMode(sensorPin, INPUT);  // กำหนดขาอินพุตเป็นโหมดอินพุต

}

void loop() {

  sensorValue = digitalRead(sensorPin);  // อ่านค่าจากเซ็นเซอร์

  if (sensorValue == HIGH) {    // ทำบางสิ่งเมื่อเซ็นเซอร์เป็นสถานะ HIGH

  }

}
```

2. ขาเอาต์พุตของ Arduino ใช้ในการส่งสัญญาณไปยังอุปกรณ์อื่นหรือควบคุมอุปกรณ์ที่ต่อกับบอร์ด จะใช้ฟังก์ชัน `digitalWrite()` เพื่อเปลี่ยนสถานะขาเอาต์พุตเป็น HIGH หรือ LOW

ตัวอย่างการควบคุมขาเอาต์พุต

```
int ledPin = 13; // กำหนดขาเอาต์พุตที่ LED ต่อเข้า

void setup() {

  pinMode(ledPin, OUTPUT); // กำหนดขาเอาต์พุตเป็นโหมดเอาต์พุต

}

void loop() {

  digitalWrite(ledPin, HIGH); // เปิด LED

  delay(1000); // รอ 1 วินาที

  digitalWrite(ledPin, LOW); // ปิด LED

  delay(1000); // รอ 1 วินาที

}
```

การควบคุมขาอินพุตและขาเอาต์พุตเป็นส่วนสำคัญในการสร้างโครงการอิเล็กทรอนิกส์ด้วย Arduino โดยสามารถนำข้อมูลจากเซ็นเซอร์หรือควบคุมอุปกรณ์อื่น ๆ ได้อย่างยืดหยุ่นและมีประสิทธิภาพ

ขาอนาล็อกอินพุต บนบอร์ด Arduino

ใช้ในการรับค่าอนาล็อกอินพุตที่เป็นค่าแรงดัน ซึ่งสามารถระบุค่าระหว่าง 0 ถึง 5 โวลต์ (V) หรือ 0 ถึง 3.3 โวลต์ (กับบอร์ดที่ใช้แรงดัน 3.3V) ในการตรวจวัดค่าความเป็นไปได้ของค่าแรงดันที่เป็นผลมาจากเซ็นเซอร์หรือแหล่งกำเนิดอื่น ๆ ที่ส่งสัญญาณไฟฟ้ามายัง Arduino ขาอนาล็อกอินพุต บ่งชี้ถึงความสามารถในการรับค่าแรงดันแบบอนาล็อกตัวอย่างขาอินพุตแบบ อนาล็อกบนบอร์ด Arduino คือ A0, A1, A2, A3, A4, และ A5 เป็นต้น ในการใช้งานขาอนาล็อกอินพุต ให้เชื่อมเซ็นเซอร์หรือแหล่งกำเนิดที่ใช้แรงดันกับขาอินพุตอนาล็อกและใช้ฟังก์ชัน `analogRead()` เพื่ออ่านค่าความเป็นไปได้ของแรงดันที่เซ็นเซอร์ส่งมา ตัวอย่างการใช้ขาอินพุตอนาล็อก

```
int sensorPin = A0; // กำหนดขาอนาล็อกอินพุต ที่เซ็นเซอร์ต่อเข้าขา A0

int sensorValue = 0; // ตัวแปรสำหรับเก็บค่าแรงดันแบบอนาล็อก

void setup() {
```

```
  // ไม่จำเป็นต้องกำหนดโหมดขาอนาล็อกอินพุต
}
void loop() {
  sensorValue = analogRead(sensorPin); // อ่านค่าแรงดันแบบอนาล็อก
}
```

ขาอนาล็อกอินพุต ช่วยให้ Arduino สามารถรับค่าที่มีความแปรปรวนและความเปลี่ยนแปลงแบบต่อเนื่องจากเซ็นเซอร์หรือแหล่งกำเนิดอื่น ๆ ที่ส่งสัญญาณแรงดันออกมา ที่สำคัญคือต้องระวังความสัมพันธ์ระหว่างค่าที่วัดและค่าจริงของของเซ็นเซอร์หรือแหล่งกำเนิดที่ใช้ เนื่องจากการเปลี่ยนแปลงความสัมพันธ์นี้อาจต้องปรับค่าหรือแปลงหน่วยให้ถูกต้องตามความต้องการ

บอร์ด Arduino มีขาอนาล็อกเอาต์พุต

ซึ่งใช้ในการส่งสัญญาณแรงดันออกมา ในการควบคุมอุปกรณ์หรือในการส่งสัญญาณแบบอนาล็อก ไปยังอุปกรณ์อื่น ๆ ที่ต่อกับบอร์ด Arduino ขาเอาต์พุตแบบ อนาล็อก สามารถสร้างสัญญาณที่มีค่าแรงดันที่เปลี่ยนแปลงต่อเนื่องในช่วงค่าหนึ่ง และค่านี้สามารถถูกควบคุมโดยโปรแกรมที่รันบนบอร์ด Arduino อาทิเช่น Arduino Uno มี 6 ขาอนาล็อกเอาต์พุต ที่เป็นตัวแปรที่ควบคุมค่าแรงดันได้แบบความแปรปรวน ซึ่งมีความสามารถระบุค่าระหว่าง 0 ถึง 5 โวลต์ (V) หรือ 0 ถึง 3.3 โวลต์ (กับบอร์ดที่ใช้แรงดัน 3.3V) โดย Arduino Uno มีขาอนาล็อกเอาต์พุต ที่ระบุด้วยตัวเลข A0, A1, A2, A3, A4, และ A5 ที่สามารถใช้สัญญาณออกแบบอนาล็อก การใช้ขาเอาต์พุตอนาล็อก สามารถใช้ฟังก์ชัน `analogWrite()` เพื่อสร้างสัญญาณแรงดันที่เปลี่ยนแปลงต่อเนื่องในช่วงค่าหนึ่ง โดยค่าที่สามารถส่งไปยังขาอนาล็อกเอาต์พุต จะอยู่ในช่วง 0 (0 โวลต์) ถึง 255 (5 โวลต์) โดยค่า 0 คือสถานะต่ำสุด (0 โวลต์) และค่า 255 คือสถานะสูงสุด (5 โวลต์) โดยมาตรฐาน

ตัวอย่างการใช้ขาอนาล็อกเอาต์พุต

```
int ledPin = 9; // กำหนดขาเอาต์พุตที่ LED ต่อเข้า
int brightness = 0; // ความสว่างของ LED
void setup() {
  // ไม่จำเป็นต้องกำหนดโหมดขาอนาล็อกเอาต์พุต
}
void loop() {
```

```
  analogWrite(ledPin, brightness); // สร้างความสว่างของ LED
  brightness = brightness + 5; // เพิ่มความสว่าง
  delay(30); // รอเวลาเล็กน้อย
  if (brightness >= 255) {
    brightness = 0; // รีเซ็ตความสว่างเมื่อถึง 255
  }
}
```

ขาอนาล็อกเอาต์พุต ช่วยให้สร้างสัญญาณแบบอนาล็อก ที่ควบคุมค่าแรงดันได้แบบต่อเนื่อง ซึ่งเหมาะสำหรับควบคุมความสว่างของ LED ควบคุมความเร็วของมอเตอร์ หรือควบคุมอุปกรณ์แบบอิเล็กทรอนิกส์ อื่นๆ ที่ต้องการควบคุมค่าแรงดันได้ในช่วงค่าหนึ่ง

สรุปท้ายบท

การเขียนโปรแกรมควบคุมขาอินพุตและเอาต์พุต แบบอนาล็อกและดิจิทัลในบอร์ด Arduino สามารถช่วยให้สามารถควบคุมและอ่านค่าจากขาอินพุต และหรือเอาต์พุต แบบอนาล็อกและดิจิทัลของบอร์ด Arduino โดยที่ผู้ออกแบบระบบ จะต้องเลือกใช้ขาอินพุต หรือขาเอาต์พุต ให้ตรงกับความต้องการว่าเป็นสัญญาณอนาล็อก หรือสัญญาณดิจิทัล เพื่อให้การประมวลผลบนบอร์ด Arduino ทำงานได้ตรงตามที่ต้องการได้

คำถามท้ายบท

1. คำสั่ง `pinMode(pin, mode)` ใช้ทำอะไร และมีโหมดใดบ้าง

2. คำสั่ง `digitalWrite(pin, value)` ใช้ทำอะไร

3. คำสั่ง `digitalRead(pin)` ใช้ทำอะไร

4. คำสั่ง `analogRead(pin)` ใช้ทำอะไร และค่าที่ได้จะอยู่ในช่วงเท่าไร

5. คำสั่ง `analogWrite(pin, value)` ใช้ทำอะไร และค่าที่ใช้ได้จะอยู่ในช่วงเท่าไร

เอกสารอ้างอิง

1. เดชฤทธิ์ มณีธรรม, *คัมภีร์การใช้งานไมโครคอนโทรลเลอร์ Arduino*, กรุงเทพฯ: ซีเอ็ดยูเคชั่น, 2560.

2. สุวิทย์ เมาะราษี, *การโปรแกรมมิ่งบอร์ด Arduino ด้วย C# .NET และ Sketch*, สำนักพัฒนาสมรรถนะครูและบุคลากรอาชีวศึกษา, 2562.

3. "*Arduino - Home*," Arduino.cc. [Online]. Available: https://www.arduino.cc/., 2023.

4. ทศพล บ้านคลองสี่, *Practical Microcontroller Programming with Arduino*, กรุงเทพฯ: ไอดีซีพรีเมียร์, 2565.

5. J. Blum, *Exploring Arduino: Tools and Techniques for Engineering Wizardry*, Hoboken, NJ: Wiley, 2013.

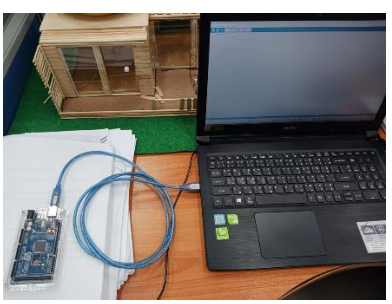
6. http://www.engineer007.com/index.php?lite-article&qid=42096260: Accessed: Jul.2, 2024

บทที่ 4 ภาคปฏิบัติในการควบคุมขาอินพุตและเอาต์พุตของ Arduino

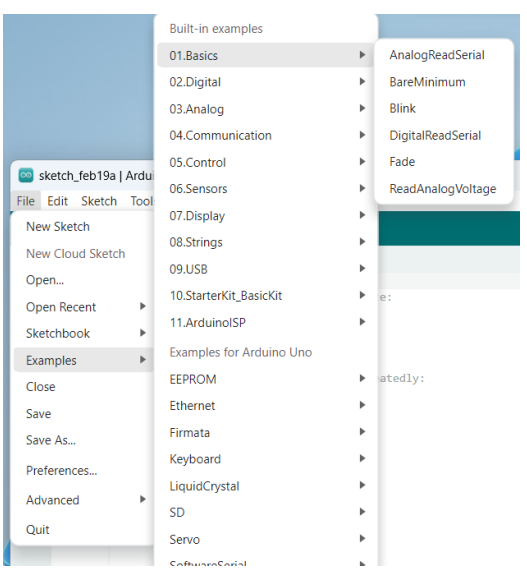
การควบคุมขาอินพุตและเอาต์พุตเป็นพื้นฐานสำคัญในการพัฒนาระบบฝังตัว (Embedded Systems) และการสร้างต้นแบบอิเล็กทรอนิกส์ (Electronic Prototyping) Arduino เป็นเครื่องมือที่มีประสิทธิภาพสำหรับการศึกษาและพัฒนาฮาร์ดแวร์ Arduino ที่มีขาดิจิทัลอินพุต/เอาต์พุต และขาอนาล็อกอินพุต/เอาต์พุต ซึ่งสามารถเชื่อมต่อกับอุปกรณ์ต่างๆ เช่น เซ็นเซอร์, มอเตอร์ และไฟ LED ได้อย่างง่ายได้ ในภาคปฏิบัติการนี้ ผู้อ่านจะได้เรียนรู้การตั้งค่าและใช้งานขาอินพุต/เอาต์พุตของ Arduino โดยใช้ Arduino IDE เพื่อเขียนโปรแกรมควบคุม การทดลองจะครอบคลุมตั้งแต่การอ่านค่าจากเซนเซอร์ การประมวลผลข้อมูล ไปจนถึงการส่งสัญญาณควบคุมไปยังอุปกรณ์ การเข้าใจและใช้งานขาอินพุตและเอาต์พุตอย่างมีประสิทธิภาพเป็นพื้นฐานสำคัญสำหรับการพัฒนาทักษะด้านระบบฝังตัวที่ใช้งานได้จริง

ตัวอย่างที่ 4.1 การเขียนโปรแกรมไฟกระพริบติดดับทุกๆ 1 วินาทีบนบอร์ด Arduino

ให้ทำการใช้สายเชื่อมระหว่างบอร์ด Arduino กับโน๊ตบุ๊คดังรูปที่ 4.1 สามารถนำโปรแกรมที่เป็นตัวอย่างได้จากเมนูดังรูปที่ 4.2

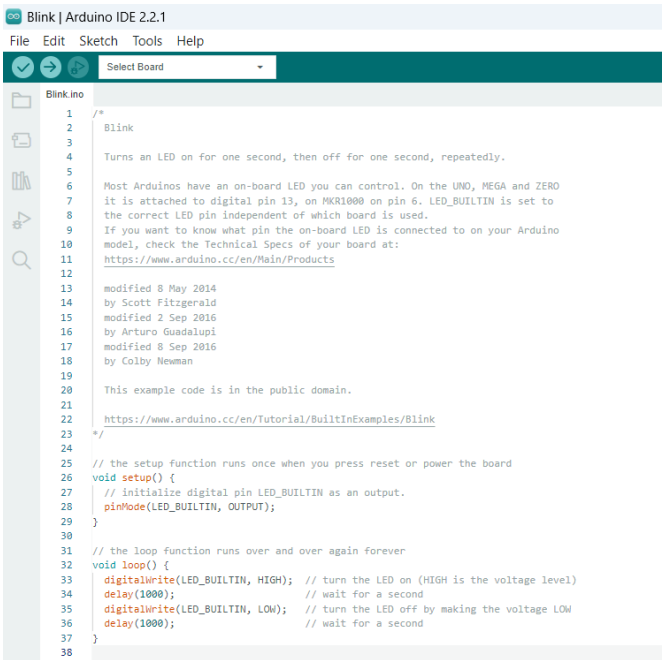


รูปที่ 4.1 การติดตั้งการเชื่อมบอร์ด Arduino กับโน๊ตบุ๊ค



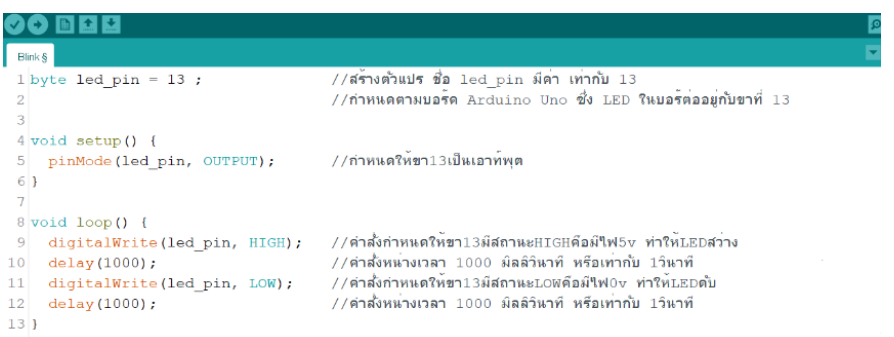
รูปที่ 4.2 เมนูสำหรับการเลือกตัวอย่างโปรแกรมที่มีในโปรแกรม Arduino IDE

ทำการเลือกโปรแกรม Blink แล้วเลือกโปรแกรมดังกล่าว ดังรูปที่ 4.3



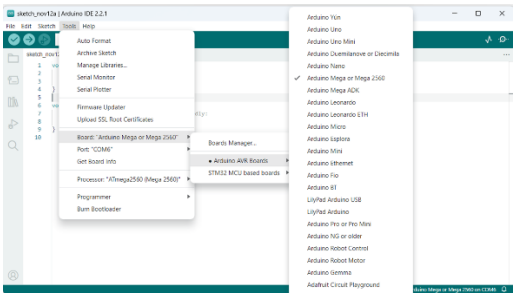
รูปที่ 4.3 การเขียนโปรแกรม Blink ที่มาจากโปรแกรมตัวอย่าง

3.ในโปรแกรมตัวอย่าง Blink จะมีข้อความที่เขียนในโปรแกรมตัวอย่าง ซึ่งสามารถเลือกเฉพาะส่วนของโปรแกรมเท่านั้น ดังรูปที่ 4.4



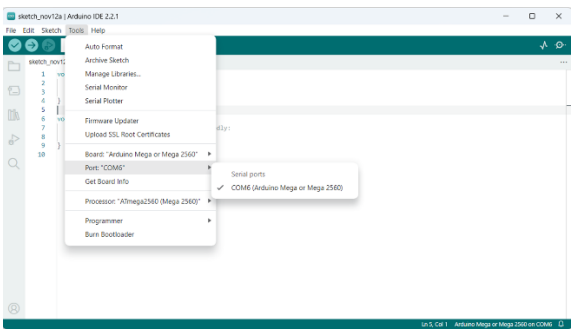
รูปที่ 4.4 การเขียนโปรแกรม Blink

4.ไปที่ Tools เพื่อเลือก board เป็นรุ่น Arduino MEGA2560 แสดงได้ดังรูปที่ 4.5



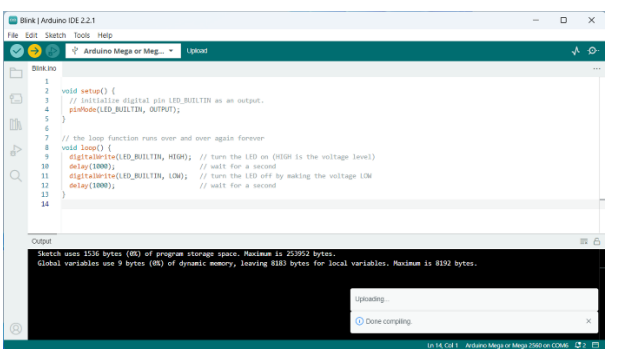
รูปที่ 4.5 เมนูในการเลือกบอร์ด Arduino ให้ตรงรุ่นที่ใช้งาน

5.ไปที่ Tools เพื่อเลือกพอร์ตในการสื่อสารระหว่างบอร์ด Arduino MEGA2560 กับโน๊ตบุ๊ค แสดงดังรูปที่ 4.6



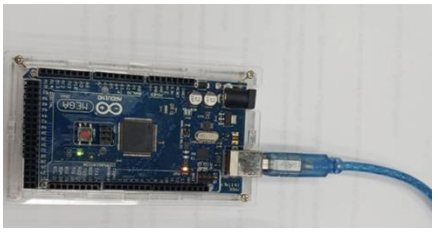
รูปที่ 4.6 เมนูในการเลือกพอร์ตที่เชื่อมต่อระหว่างบอร์ด Arduino กับคอมพิวเตอร์

6.กดอัปโหลดโปรแกรม เมื่ออัปโหลดเสร็จสิ้นแสดงข้อความ Done uploading แสดงได้ดังรูปที่ 4.7



รูปที่ 4.7 ผลลัพธ์ เมื่ออัปโหลดโปรแกรมลงในบอร์ด Arduino เรียบร้อย

7.ผลการทำงานโปรแกรม จะสังเกต led บนบอร์ด Arduino MEGA2560 ที่จะกระพริบติดดับทุกๆ 1วินาที แสดงได้ดังรูปที่ 4.8

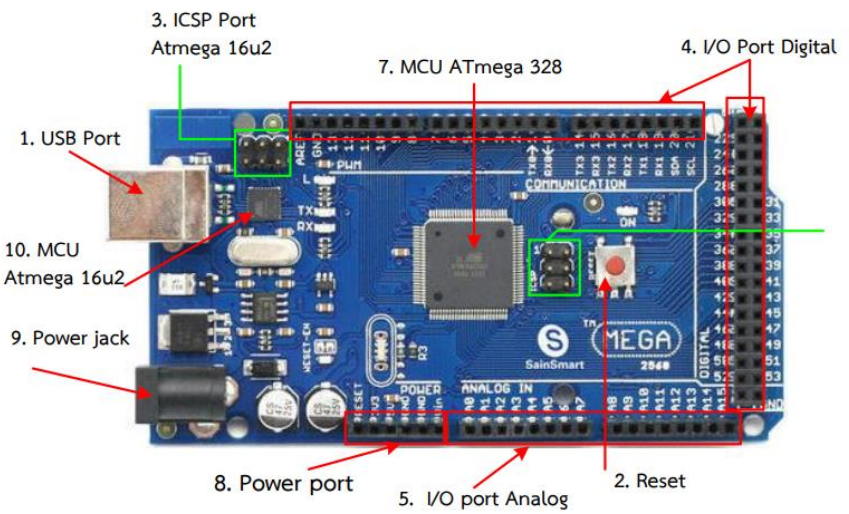


รูปที่ 4.8 ผลลัพธ์ว่าไฟ LED บนบอร์ด Arduino MEGA2560 ที่จะติดและดับทุกๆ 1วินาที

4.1 โครงสร้างบอร์ด Arduino MEGA2560

ในโครงสร้างบอร์ด แสดงได้ดังรูปที่ 4.9 จะประกอบด้วย ดังต่อไปนี้

1. พอร์ตยูเอสบี ใช้สำหรับต่อกับ Computer เพื่ออับโหลดโปรแกรมเข้า MCU และจ่ายไฟให้กับบอร์ด
2. ปุ่มเริ่มต้นใหม่ เป็นปุ่มเริ่มต้นใหม่ ใช้กดเมื่อต้องการให้MCU เริ่มการทำงานใหม่



รูปที่ 4.9 โครงสร้างบอร์ด Arduino MEGA2560

3. ICSP พอร์ต ของ Atmega16U2 เป็นพอร์ตที่ใช้โปรแกรม Visual Com port บน Atmega16U2
4. ดิจิทัลอินพุต/เอาต์พุตพอร์ต ตั้งแต่ขา D0 ถึง D54
5. อนาล็อกอินพุต/เอาต์พุตพอร์ต ตั้งแต่ขา A0ถึง A15
6. ICSP พอร์ต Atmega328 เป็นพอร์ตที่ใช้โปรแกรม Bootloader
7. MCU Atmega328 เป็น MCU ที่ใช้บนบอร์ด Arduino
8. Power Port ไฟเลี้ยงของบอร์ดเมื่อต้องการจ่ายไฟให้กับวงจรภายนอก ประกอบด้วยขาไฟเลี้ยง + 3.3 V, +5V, GND, Vin

9. Power Jack รับไฟจาก Adapter โดยที่แรงดันอยู่ระหว่าง 7-12 V

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

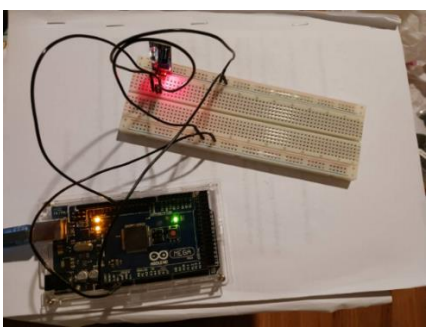
4.2 การใช้งานบอร์ด Arduino MEGA2560 กับเซ็นเซอร์อินฟราเรด (IR Sensor)

4.2.1 การเชื่อมโยงอุปกรณ์ที่ใช้

การเชื่อมโยงสายระหว่างเซ็นเซอร์อินฟราเรด กับบอร์ด Arduino MEGA2560 จะทำการติดตั้งขาต่างๆ แสดงได้ดังรูปที่ 4.10

รูปที่ 4.10 การติดตั้งสายระหว่างบอร์ด Arduino กับเซ็นเซอร์อินฟราเรด

และเมื่อนำอุปกรณ์มาต่อกัน จะแสดงได้ดังรูปที่ 4.11



รูปที่ 4.11 การต่อวงจรจริง ระหว่างบอร์ด Arduino กับ เซ็นเซอร์อินฟราเรด

4.2.2 การเขียนโปรแกรม

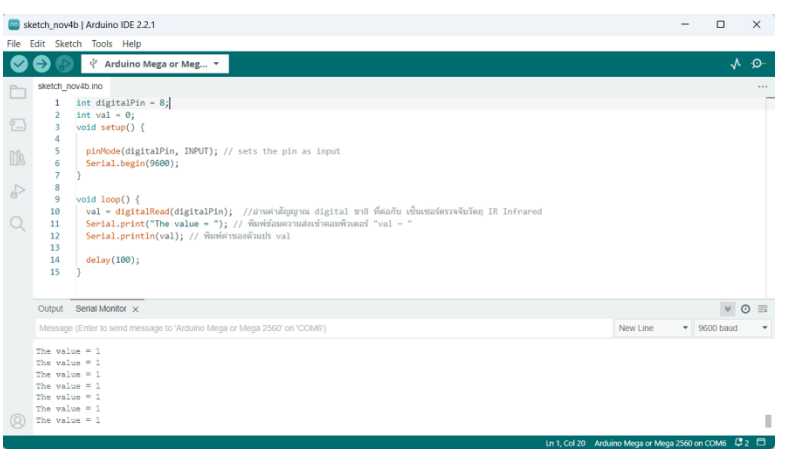
ในส่วนที่เป็นการเขียนโปรแกรม จะใช้การเขียนโปรแกรมด้วยคำสั่งดังนี้

```
int digitalPin = 8; /* ประกาศตัวแปร digitalPin และกำหนดให้มีค่าเท่ากับ 8 ซึ่งเป็นหมายเลขของขาดิจิทัล
ที่เชื่อมต่อกับเซ็นเซอร์*/
```

```
int val = 0; /* ประกาศตัวแปร val และกำหนดค่าเริ่มต้นให้เป็น 0 เพื่อใช้ในการเก็บค่าสัญญาณที่อ่านได้จาก digitalPin*/
void setup( ) {
  pinMode(digitalPin, INPUT); /* กำหนดให้ขา digitalPin ทำหน้าที่เป็นขาอินพุตเพื่อรับค่าสัญญาณจากเซ็นเซอร์*/
  Serial.begin(9600); /* เริ่มต้นการสื่อสารแบบ Serial ที่อัตราความเร็ว 9600 bps (บิตต่อวินาที) เพื่อส่งข้อมูลไปยังคอมพิวเตอร์*/
}
void loop( ) {
  val = digitalRead(digitalPin); /*อ่านค่าสัญญาณดิจิทัลจากขา digitalPin (ขา 8) และเก็บค่าไว้ในตัวแปร val*/
  Serial.print("The value = "); // พิมพ์ข้อความ "The value = " ไปยังคอมพิวเตอร์ผ่าน Serial
  Serial.println(val); // พิมพ์ค่าของตัวแปร val ไปยังคอมพิวเตอร์และขึ้นบรรทัดใหม่
  delay(1000); /* หยุดการทำงานของโปรแกรมเป็นเวลา 1000 มิลลิวินาที (1 วินาที) เพื่อทำให้การอ่านค่าและแสดงผลเกิดขึ้นทุกๆ วินาที*/
}
```

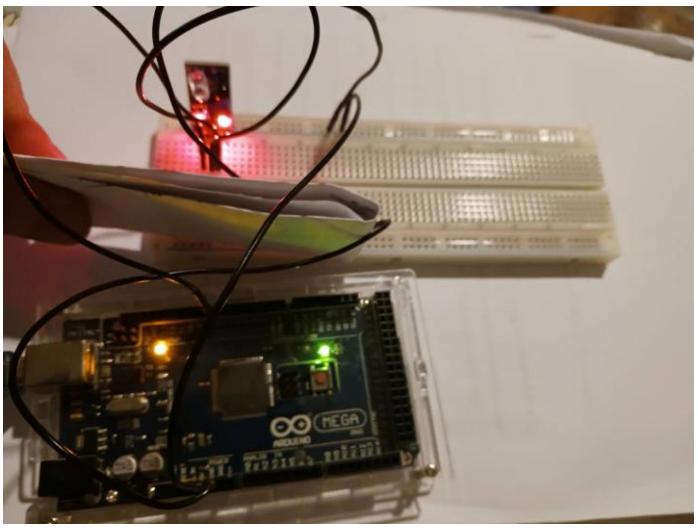
ให้ใส่โปรแกรมข้างต้น ใส่ในโปรแกรม Arduino IDE แล้วทำการอัปโหลดเข้าบอร์ด Arduino MEGA2560 พอทำสำเร็จ ให้ทำการทดสอบการตรวจจับวัตถุของเซ็นเซอร์อินฟราเรด โดยที่หากไม่มีวัตถุอยู่ด้านหน้า เซ็นเซอร์อินฟราเรด จะมีทำให้เกิดผลลัพธ์เป็น 1 แสดงได้ดังรูปที่ 4.12

4.2.3 ผลลัพธ์ที่ได้

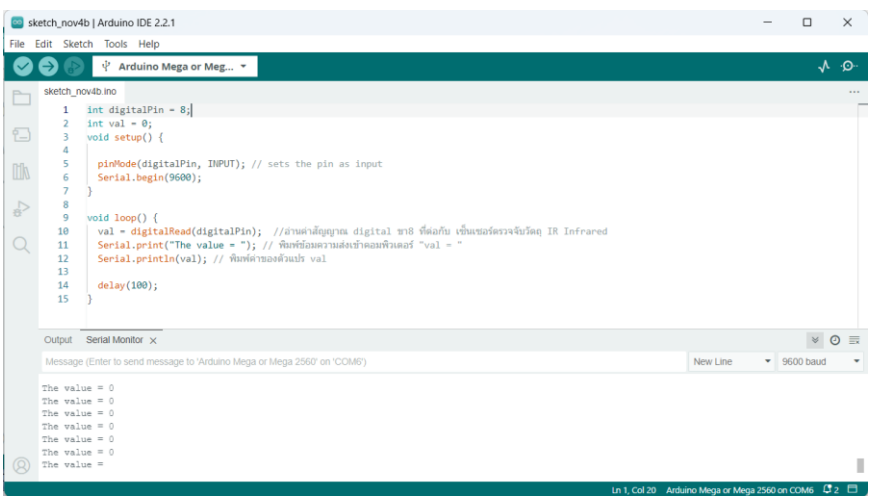


รูปที่ 4.12 ผลลัพธ์ขณะยังไม่มีอะไรมากั้นเซ็นเซอร์อินฟราเรด

ทำนองเดียวกัน หากมีการนำกระดาษมากั้นที่ IR Sensor แสดงดังรูปที่ 4.13 แล้วผลลัพธ์ที่ได้จะมีค่าเป็น 0 แสดงได้ดังรูปที่ 4.14



รูปที่ 4.13 การเอากระดาษมากั้นเซ็นเซอร์อินฟราเรด



รูปที่ 4.14 ผลลัพธ์ขณะมีอะไรมากั้นเซ็นเซอร์อินฟราเรด

4.3 การใช้งานบอร์ด Arduino กับเซ็นเซอร์ DHT วัดความชื้นและอุณหภูมิ

DHT11 และ DHT22 คือเซ็นเซอร์วัดความชื้นและอุณหภูมิในอากาศ สามารถนำไปประยุกต์ใช้งานได้หลากหลาย เช่น ตู้ฟักไข่ โรงเรือนปลูกผัก โรงเพาะเห็ด เป็นต้น นำไปใช้งานได้ง่าย รองรับแรงดันไฟฟ้า 3.3 V และ 5V

4.3.1 เซ็นเซอร์วัดความชื้นและอุณหภูมิ

4.3.1.1 DHT11

เป็นเซ็นเซอร์ที่มีคุณสมบัติได้ดังนี้

- อุณหภูมิ 0 ถึง 50 °C ความผิดพลาด ±2 °C
- ความชื้น 20 ถึง 80 % ความผิดพลาด ±5 %

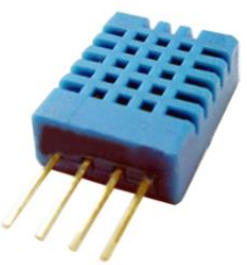
ซึ่งเซ็นเซอร์ DHT11 จะมีรูปแบบต่างๆ ดังรูปที่ 4.15

4.3.1.2 DHT22

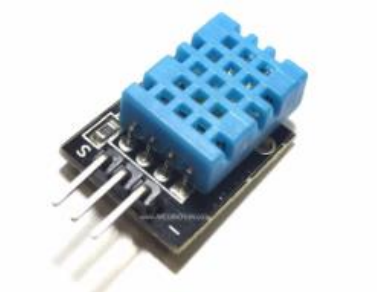
เป็นเซ็นเซอร์ที่มีคุณสมบัติได้ดังนี้

- อุณหภูมิ -40 ถึง 80 °C ความผิดพลาด ±0.5 °C
- ความชื้น 0 ถึง 100 % ความผิดพลาด ±5 %

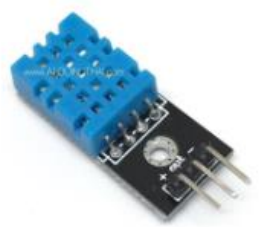
ซึ่งเซ็นเซอร์ DHT22 จะมีรูปแบบต่างๆ ดังรูปที่ 4.16



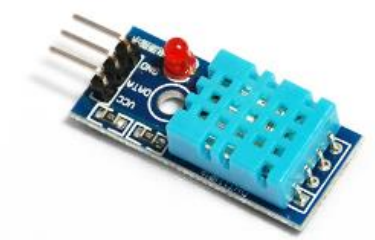
โมดูล DHT11 แบบธรรมดา



โมดูล DHT11



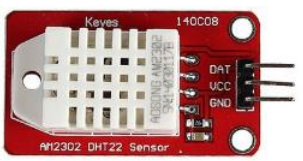
โมดูล DHT11



โมดูล DHT11 พร้อม LED

รูปที่ 4.15 เซ็นเซอร์วัดความชื้นและอุณหภูมิ ชนิดโมดูล DHT11

DHT22 แบบธรรมดา



โมดูล DHT22 บอร์ดแดง

โมดูล DHT22

โมดูล DHT22 AM2302

รูปที่ 4.16 เซ็นเซอร์วัดความชื้นและอุณหภูมิ ชนิดโมดูล DHT22

4.3.2 การเชื่อมโยงอุปกรณ์ที่ใช้

รายการอุปกรณ์

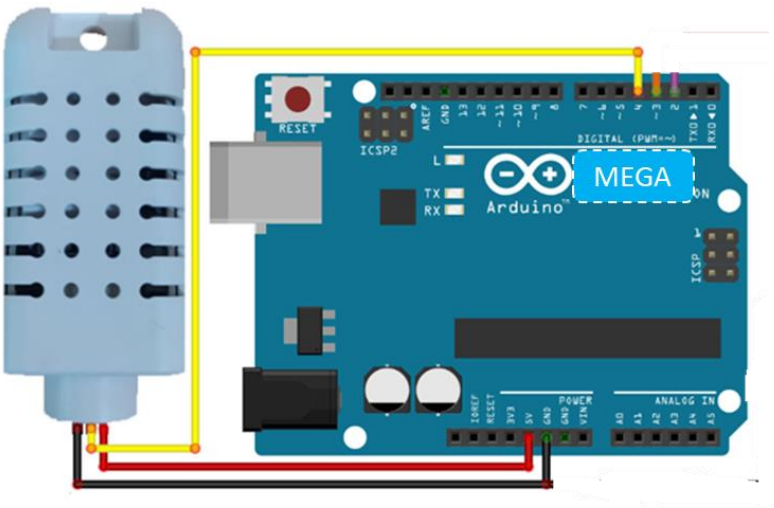
- Arduino MEGA2560
- โมดูล DHT22 AM2302
- สายจั้มเปอร์

DHT22 AM2302 ----> Arduino MEGA2560

ให้ทำการเชื่อมโยงโดยเรียงตามลำดับ ระหว่างสายสีต่างๆ จากเซ็นเซอร์ต่อเข้ากับขา Arduino MEGA2560 ดังนี้

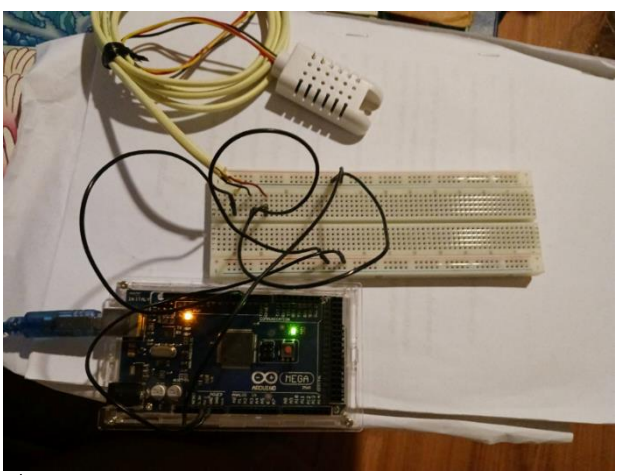
- VCC(แดง) ----> ขา 5V
- GND(ดำ) ----> ขา GND
- DATA(เหลือง) ----> ขา 4

ซึ่งแสดงได้ดังรูปที่ 4.17



รูปที่ 4.17 การต่อสายระหว่างบอร์ด Arduino MEGA2560 กับ DHT22

เมื่อทำการต่อเซ็นเซอร์ DHT22 กับบอร์ด Arduino MEGA2560 ผ่านบอร์ดวงจรที่แสดงให้เห็นการต่อวงจรจริงได้ดังรูปที่ 4.18



รูปที่ 4.18 การต่อวงจรจริง ระหว่างบอร์ด Arduino กับ DHT22

ถัดจากนี้ ให้ทำการเขียนโปรแกรมเพื่อเตรียมอัปโหลดลงบอร์ด Arduino ดังหัวข้อถัดไป ซึ่งก่อนจะเขียนโปรแกรม จะต้องทำการติดตั้งไลบรารี DHT22 ให้เห็นในโปรแกรม Arduino IDE แสดงได้ดังรูปที่ 4.19 และให้เห็นว่าพอร์ตที่เชื่อมโยงระหว่างบอร์ด Arduino กับคอมพิวเตอร์ แสดงได้ดังรูปที่ 4.20 เรียบร้อย แล้วจึงค่อยเขียนโปรแกรมต่อไป

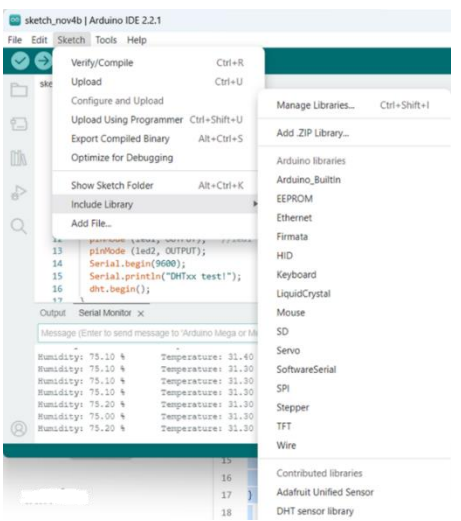
4.3.3 แสดงการเขียนโปรแกรม

ให้เขียนโปรแกรมดังนี้

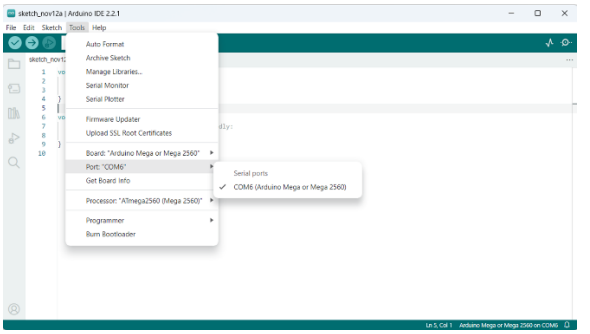
```
#include <DHT.h>// เรียกใช้ไลบรารี DHT
#define DHTPIN 4 /* กำหนดขาข้อมูลที่เซ็นเซอร์ DHT22 ต่อเข้ากับขา 4*/
#define DHTTYPE DHT22   /* กำหนดชนิดของเซ็นเซอร์ DHT ที่ใช้งานเป็น DHT22*/
float tempsetting = 30;  /*กำหนดค่าอุณหภูมิที่ต้องการให้คำสั่งทำงาน ซึ่งสามารถเปลี่ยนแปลงได้ตามความต้องการ*/
DHT dht(DHTPIN, DHTTYPE); /* สร้างอ็อบเจกต์เซ็นเซอร์ DHT โดยกำหนดขาข้อมูลและชนิดของเซ็นเซอร์*/
void setup()
{
  Serial.begin(9600); /* เริ่มต้นการสื่อสารที่อัตราสัญญาณ 9600 bps*/
  Serial.println("DHTxx test!"); /* พิมพ์ข้อความ "DHTxx test!" ไปยังหน้าจอคอมพิวเตอร์ผ่าน Serial Monitor*/
  dht.begin();// เริ่มต้นการใช้งานเซ็นเซอร์ DHT
}
void loop() {
  delay(2000); // หยุดการทำงานของโปรแกรมเป็นเวลา 2 วินาที
  float h = dht.readHumidity();/* อ่านค่าความชื้นจากเซนเซอร์ DHT*/
  float t = dht.readTemperature();/* อ่านค่าอุณหภูมิในหน่วยเซลเซียสจากเซนเซอร์ DHT*/
  float f = dht.readTemperature(true); /* อ่านค่าอุณหภูมิในหน่วยฟาเรนไฮต์จากเซนเซอร์ DHT*/
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
```

```
  Serial.print("Humidity: ");/* พิมพ์ข้อความ "Humidity: " ไปยังหน้าจอ*/
  Serial.print(h); // พิมพ์ค่าความชื้นที่อ่านได้จากเซนเซอร์ DHT
  Serial.println(" %\t"); // พิมพ์หน่วย % และขึ้นบรรทัดใหม่
  Serial.print("Temperature: ");/* พิมพ์ข้อความ "Temperature: " ไปยังหน้าจอ*/
  Serial.print(t); /* พิมพ์ค่าอุณหภูมิในหน่วยเซลเซียสที่อ่านได้จากเซนเซอร์ DHT*/
  Serial.println(" *C ");/* พิมพ์หน่วยองศาเซลเซียส และขึ้นบรรทัดใหม่*/
}
```

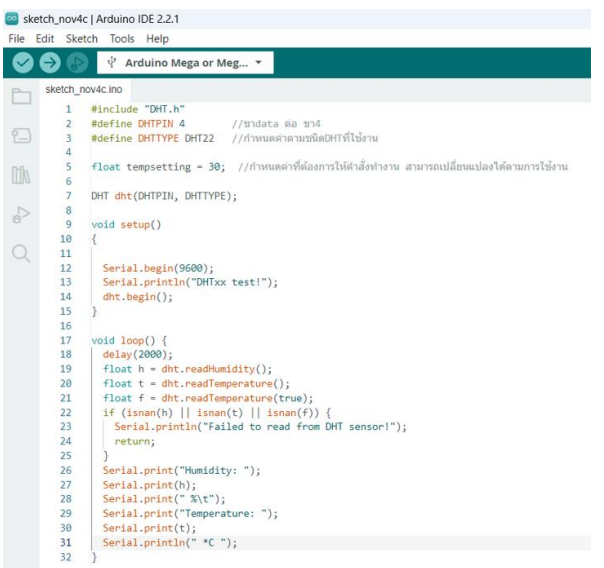
แล้วเขียนโปรแกรมลงในโปรแกรม Arduino IDE ดังรูปที่ 4.21



รูปที่ 4.19 ผลลัพธ์ เมื่อ DHT22 ถูกติดตั้งเรียบร้อย



รูปที่ 4.20 เมนูพอร์ตที่เชื่อมติดต่อระหว่าง บอร์ด Arduino MEGA 2560 กับคอมพิวเตอร์



รูปที่ 4.21 การเขียนโปรแกรมลงในโปรแกรม Arduino IDE

ถัดจากนั้น ให้ทำการอัปโหลดโปรแกรมลงบอร์ด Arduino MEGA2560 แล้วจะแสดงผลลัพธ์ในหัวข้อถัดไป

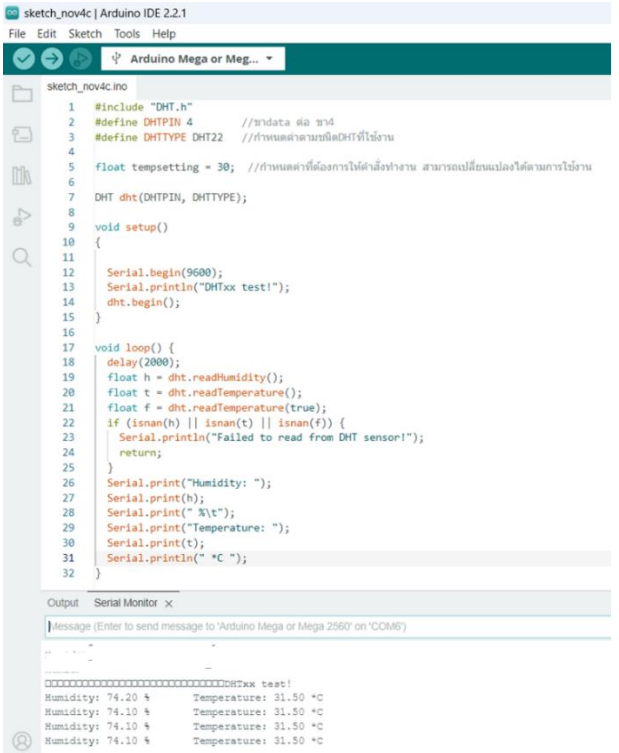
4.3.4 ผลลัพธ์ที่ได้

4.3.4.1 ผลการทดสอบการวัดความชื้นและอุณหภูมิห้อง

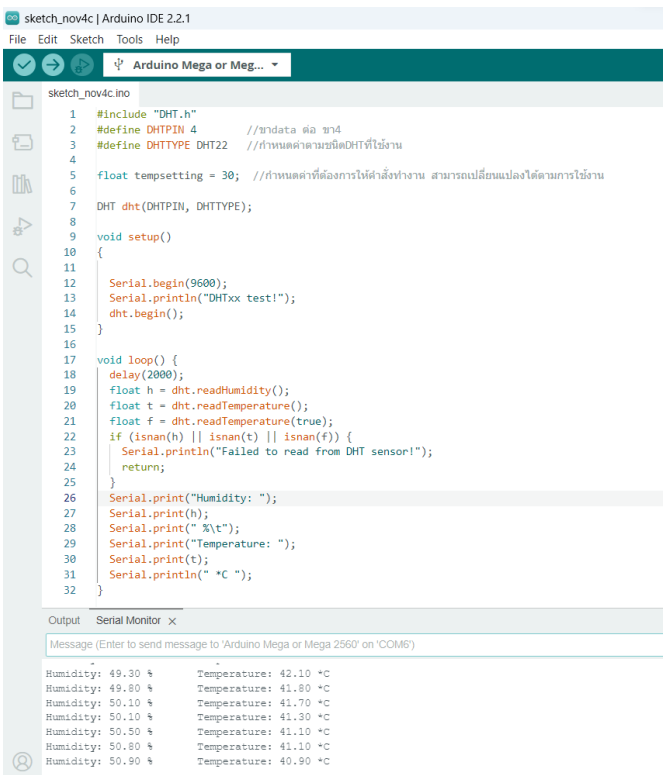
ผลการทดสอบจะแสดงผลการวัดความชื้น และอุณหภูมิห้องได้ดังรูปที่ 4.22

4.3.4.2 ผลการทดสอบการวัดความชื้นและอุณหภูมิห้อง เมื่อมีการเพิ่มค่าความร้อนจากไดร์เป่าผม

พอใช้ไดร์เป่าผม เป่าไปที่เซ็นเซอร์ DHT22 จะทำให้อุณหภูมิสูงขึ้น ทำให้ผลการทดสอบที่ได้แสดงบนหน้าจอ Serial Monitor ดังรูปที่ 4.23



รูปที่ 4.22 ผลลัพธ์การวัดความชื้น และอุณหภูมิของอากาศที่ตรวจวัดได้จาก DHT22

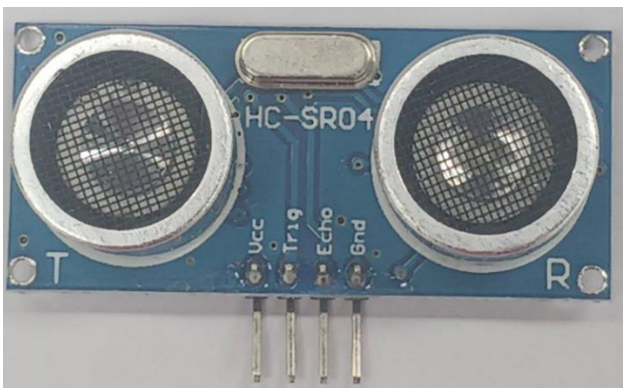


รูปที่ 4.23 ผลลัพธ์อุณหภูมิที่สูงขึ้น เมื่อใช้ไดร์เป่าผมเพิ่มอุณหภูมิให้ DHT22

4.4 การควบคุมเซ็นเซอร์เหนือเสียงผ่านบอร์ด Arduino สำหรับการวัดระยะทาง

4.4.1 การเชื่อมโยงบอร์ด และอุปกรณ์ต่างๆ

จะเห็นว่า เซ็นเซอร์เหนือเสียง Module HC-SR04 จะแสดงได้ดังรูปที่ 4.24

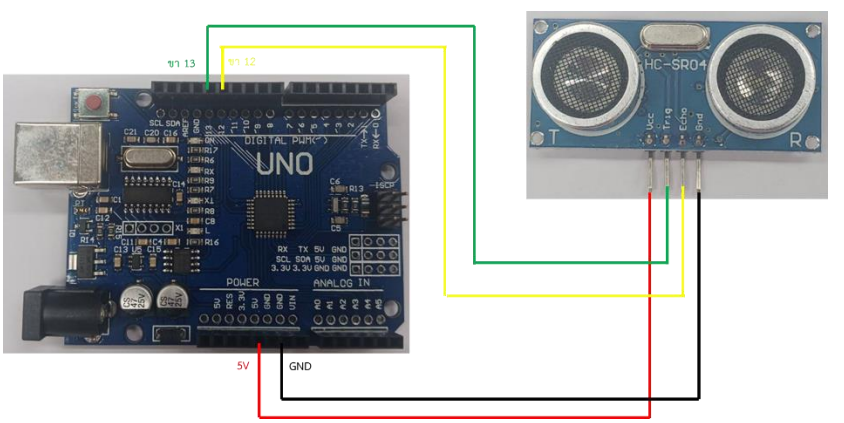


รูปที่ 4.24 เซ็นเซอร์เหนือเสียง Module HC-SR04

วิธีการต่ออุปกรณ์

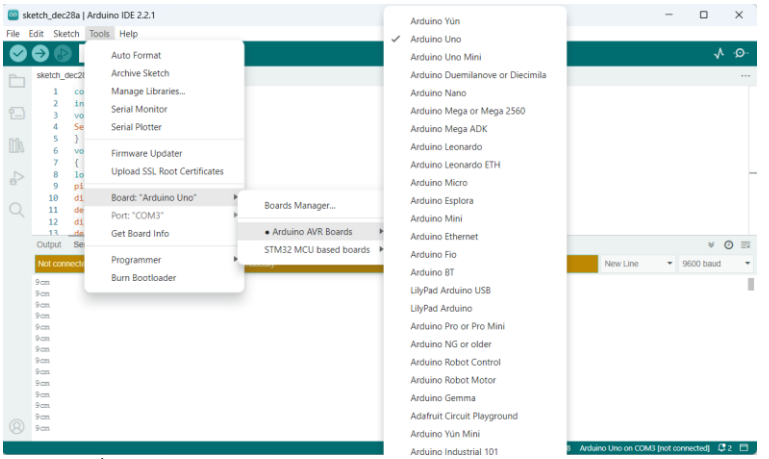
ให้ทำการต่อสายไฟระหว่างบอร์ด Arduino UNO R3 กับ เซ็นเซอร์เหนือเสียง ดังรูปที่ 4.25 ดังนี้

- ให้ขา12 ของบอร์ด Arduino ต่อ -> ขา Echo ของเซ็นเซอร์
- ให้ขา13 ของบอร์ด Arduino ต่อที่ -> ขา Trig ของเซ็นเซอร์
- ให้ขา5V ของบอร์ด Arduino ต่อที่ -> ขา Vcc ของเซ็นเซอร์
- ให้ขาGND ของบอร์ด Arduino ต่อที่ -> ขา GND ของเซ็นเซอร์

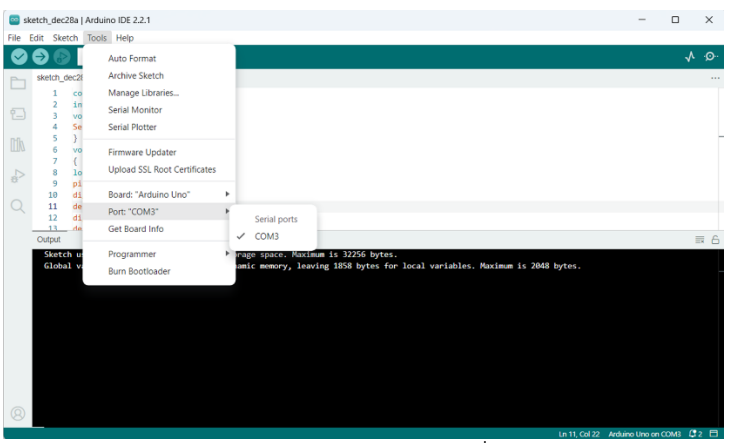


รูปที่ 4.25 การต่อเซ็นเซอร์เหนือเสียง กับบอร์ด Arduino UNO

ก่อนเขียนโปรแกรมลงในโปรแกรม Arduino IDE ต้องตรวจสอบว่าบอร์ด Arduino UNO ดังรูปที่ 4.26 สามารถสื่อสารผ่านพอร์ตของคอมพิวเตอร์ พอร์ตอะไรก่อนดังรูปที่ 4.27 เพื่อให้โปรแกรมสามารถสื่อสารระหว่างบอร์ด Arduino UNO กับ คอมพิวเตอร์ได้ก่อนเสมอ



รูปที่ 4.26 การเลือกบอร์ด Arduino UNO ใน Board List



รูปที่ 4.27 การเลือก Port COM3 สำหรับการติดต่อสื่อสารกับบอร์ด Arduino UNO

4.4.2 การเขียนโปรแกรม

ให้ทำการเขียนโปรแกรมดังนี้

```
const int pingPin = 13; // กำหนดขา pingPin เป็นขา 13 เพื่อใช้ส่งสัญญาณคลื่นเสียง (ส่ง Trig)
int inPin = 12; // กำหนดขา inPin เป็นขา 12 เพื่อรับสัญญาณสะท้อนกลับ (รับ Echo)
void setup() {
Serial.begin(9600); // เริ่มต้นการสื่อสารผ่าน Serial ที่อัตราความเร็ว 9600 bps
}
void loop()
{
long duration, cm; // ประกาศตัวแปร duration สำหรับเก็บเวลาและ cm สำหรับเก็บระยะทาง
pinMode(pingPin, OUTPUT); // ตั้งค่าให้ขา pingPin ทำหน้าที่เป็นขาส่งสัญญาณ (OUTPUT)
digitalWrite(pingPin, LOW); // เริ่มต้นส่งสัญญาณโดยตั้งค่าให้ขา pingPin เป็น LOW (ปิด)
delayMicroseconds(2); // รอ 2 ไมโครวินาที
```

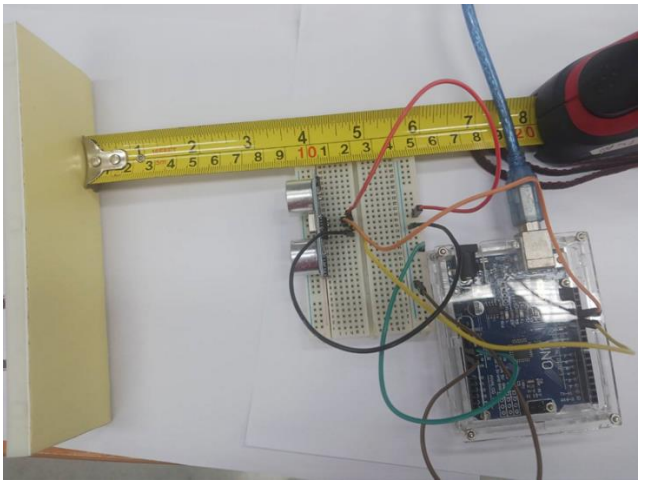
```
digitalWrite(pingPin, HIGH); // ส่งสัญญาณสั้นๆ โดยตั้งค่าให้ขา pingPin เป็น HIGH (เปิด)
delayMicroseconds(5); // รอ 5 ไมโครวินาทีเพื่อให้สัญญาณส่งออกไป
digitalWrite(pingPin, LOW); // ปิดสัญญาณหลังจากส่งออกไป
pinMode(inPin, INPUT); // ตั้งค่าให้ขา inPin ทำหน้าที่เป็นขารับสัญญาณ (INPUT)

duration = pulseIn(inPin, HIGH); /* อ่านเวลาที่สัญญาณสะท้อนกลับมาถึงขา inPin และเก็บค่าไว้ในตัวแปร duration*/

cm = microsecondsToCentimeters(duration); /* คำนวณระยะทางจากเวลาที่วัดได้โดยใช้ฟังก์ชัน microsecondsToCentimeters() */

Serial.print(cm); // พิมพ์ค่าระยะทางที่คำนวณได้ในหน่วยเซนติเมตร
Serial.print("cm");// พิมพ์หน่วย "cm" ตามหลังค่าระยะทาง
Serial.println();// ขึ้นบรรทัดใหม่
delay(100); // รอ 100 มิลลิวินาที (0.1 วินาที) ก่อนจะทำการวัดรอบต่อไป
}
long microsecondsToCentimeters(long microseconds)
{
/* ความเร็วเสียงในอากาศคือ 340 เมตรต่อวินาที หรือประมาณ 29 ไมโครวินาทีต่อเซนติเมตร

  คลื่นเสียงเดินทางไป-กลับ ดังนั้นระยะทางที่วัดได้ต้องหารด้วย 2 */

return microseconds / 29 / 2; // คืนค่าระยะทางในหน่วยเซนติเมตร
}
```

ให้เขียนโปรแกรมข้างต้น ลงในโปรแกรม Arduino IDE แล้วให้ทำการอัปโหลดโปรแกรมลงบอร์ด Arduino UNO ก็จะสามารถทำงานได้เรียบร้อย ผลที่ได้จะแสดงดังหัวข้อถัดไป

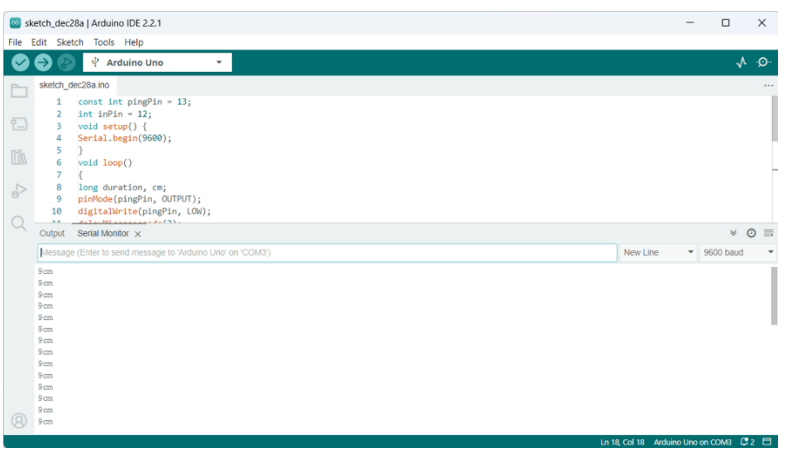
4.4.3 ผลลัพธ์ที่ได้

การทดสอบโดยการตั้งฉากกั้นที่ด้านหน้าของเซ็นเซอร์ โดยที่จะมีตลับวัดระยะ เพื่อเปรียบเทียบว่าผลที่ได้จากการตรวจจับระยะทาง ตรงกันกับความเป็นจริงหรือไม่ ซึ่งแสดงวงจรที่ติดตั้งจริงได้ดังรูปที่ 4.28

ผลการตรวจวัดระยะทาง จะดูได้ที่ Serial Monitor ซึ่งจะแสดงได้ดังรูปที่ 4.29 ผลที่ได้จะได้ระยะ 9 cm ตรงกับระยะที่กั้นหน้าเซ็นเซอร์ จึงแสดงให้เห็นว่าการใช้ Arduino UNO ควบคุมระยะทางโดยการใช้เซ็นเซอร์เหนือเสียง มีความถูกต้อง และเที่ยงตรงตามที่ได้ออกแบบโปรแกรมไว้



รูปที่ 4.28 การนำวัตถุมากั้นเซ็นเซอร์เหนือเสียง ที่ระยะทาง 9 cm

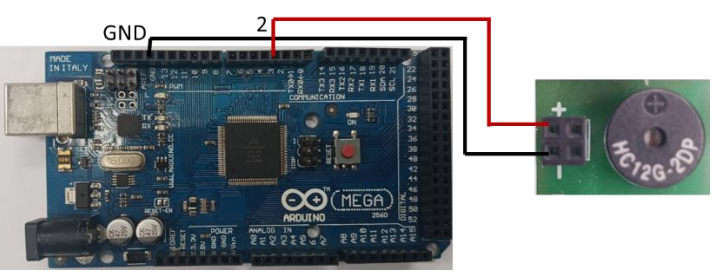


รูปที่ 4.29 ผลลัพธ์การวัดระยะทางของวัตถุกั้นเซ็นเซอร์บน Serial Monitor

4.5 การควบคุมลำโพง Piezo ผ่านบอร์ด Arduino

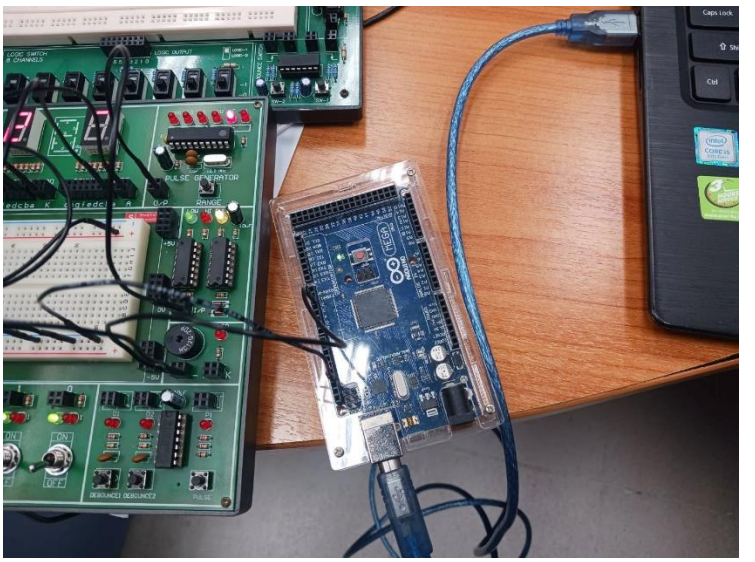
4.5.1 การเชื่อมโยงบอร์ดและอุปกรณ์

ให้ทำการต่อสายไฟระหว่างบอร์ด Arduino MEGA2560 กับ Piezo ดังรูปที่ 4.30



รูปที่ 4.30 การต่อสายไฟระหว่างบอร์ด Arduino กับ Piezo

ในการต่อวงจรจริง ผู้เขียนขอใช้เซ็นเซอร์ Piezo จากบอร์ดวงจรดิจิทัล ซึ่งจะแสดงการต่อวงจรจริงได้ดังรูปที่ 4.31



รูปที่ 4.31 การต่อวงจรจริงระหว่าง บอร์ด Arduino กับ Piezo

4.5.2 การเขียนโปรแกรม

ให้ทำการเขียนโปรแกรมดังนี้

```
int c=262; // ความถี่ของโน้ต C (Do) ในหน่วยเฮิรตซ์ (Hz)
int d =294; // ความถี่ของโน้ต D (Re) ในหน่วยเฮิรตซ์ (Hz)
int e = 330; // ความถี่ของโน้ต E (Mi) ในหน่วยเฮิรตซ์ (Hz)

void setup() {
  // ฟังก์ชัน setup() ทำงานครั้งเดียวเมื่อเริ่มต้นการทำงานของโปรแกรม
pinMode(2,OUTPUT); // กำหนดให้ขา 2 เป็นขาส่งสัญญาณเอาต์พุต (OUTPUT)
}
void loop() {
  // ฟังก์ชัน loop() จะทำงานซ้ำไปเรื่อยๆ ตลอดเวลาที่บอร์ด Arduino ทำงาน
tone(2, c, 2000); // สร้างเสียงที่ความถี่ 262 Hz (โน้ต C) ที่ขา 2 เป็นเวลา 2000 มิลลิวินาที (2 วินาที)
delay(200); // รอ 200 มิลลิวินาที (0.2 วินาที)

tone(2, d,500); // สร้างเสียงที่ความถี่ 294 Hz (โน้ต D) ที่ขา 2 เป็นเวลา 500 มิลลิวินาที (0.5 วินาที)

delay(500); // รอ 500 มิลลิวินาที (0.5 วินาที)
tone(2,e, 500); // สร้างเสียงที่ความถี่ 330 Hz (โน้ต E) ที่ขา 2 เป็นเวลา 500 มิลลิวินาที (0.5 วินาที)
```

```
delay(500); // รอ 500 มิลลิวินาที (0.5 วินาที)
}
```

ซึ่งจะทำการเขียนโปรแกรมลงบนในโปรแกรม Arduino IDE ดังรูปที่ 4.32 ถัดจากนั้น ให้ทำการอัปโหลดเข้าบอร์ด Arduino MEGA2560 เพื่อให้บอร์ด Arduino ควบคุมเสียงให้กับ Piezo

รูปที่ 4.32 โปรแกรมที่เตรียมอัปโหลดให้บอร์ด Arduino

4.5.3 ผลลัพธ์ที่ได้

การทดสอบครั้งนี้ จะเป็นเสียงโน๊ตดนตรี ที่กำหนดในโปรแกรม จะมีเสียงโด เร มี

สรุปท้ายบท

ในภาคปฏิบัติการนี้ จะได้เรียนรู้การใช้บอร์ด Arduino เพื่อควบคุมและตรวจสอบขาอินพุต/เอาต์พุต ผ่านการเขียนโปรแกรมด้วย Arduino IDE การทดลองได้ครอบคลุมตั้งแต่การอ่านค่าจากเซ็นเซอร์อนาล็อกและดิจิทัล การประมวลผลข้อมูล ไปจนถึงการส่งสัญญาณควบคุมอุปกรณ์ต่างๆ เช่น LED และเซ็นเซอร์ เป็นต้น จะได้ฝึกการตั้งค่าและใช้งานขาดิจิทัลและอนาล็อกของ Arduino ทำให้สามารถเชื่อมต่อและควบคุม

อุปกรณ์ภายนอกได้อย่างมีประสิทธิภาพ ภาคปฏิบัติการนี้ยังช่วยเสริมทักษะการวิเคราะห์และแก้ไขปัญหาเมื่อต้องทำงานกับฮาร์ดแวร์และซอฟต์แวร์ ความเข้าใจในการใช้งานขาอินพุตและเอาต์พุตของ Arduino เป็นพื้นฐานสำคัญสำหรับการพัฒนาระบบฝังตัวที่ซับซ้อนและโครงการที่ใช้งานได้จริง

คำถามท้ายบท

1. ให้แสดงการต่อวงจรระหว่างบอร์ด Arduino กับเซ็นเซอร์อินฟราเรด พร้อมเขียนโปรแกรม และอธิบายพอสังเขป

2. ให้แสดงการต่อวงจรระหว่างบอร์ด Arduino กับ เซ็นเซอร์ DHT22 พร้อมเขียนโปรแกรม และอธิบายพอสังเขป

3. ให้แสดงการต่อวงจรระหว่างบอร์ด Arduino กับ เซ็นเซอร์เหนือเสียง พร้อมเขียนโปรแกรม และอธิบายพอสังเขป

4. ให้แสดงการต่อวงจรระหว่างบอร์ด Arduino กับ Piezo พร้อมเขียนโปรแกรม และอธิบายพอสังเขป

5. ให้แสดงการนำโปรแกรมตัวอย่าง ชื่อ Blink มาใช้ในการควบคุม LED บนบอร์ด Arduino พร้อมอธิบายโปรแกรมพอสังเขป

เอกสารอ้างอิง

1. เดชฤทธิ์ มณีธรรม, *คัมภีร์การใช้งานไมโครคอนโทรลเลอร์ Arduino*, กรุงเทพฯ: ซีเอ็ดยูเคชั่น, 2560.

2. สุวิทย์ เมาะราษี, *การโปรแกรมมิ่งบอร์ด Arduino ด้วย C# .NET และ Sketch*, สำนักพัฒนาสมรรถนะครูและบุคลากรอาชีวศึกษา, ไม่ระบุปี.

3. "*Arduino - Home*," Arduino.cc. [Online]. Available: https://www.arduino.cc/. ,2023.

4. J. Blum, *Exploring Arduino: Tools and Techniques for Engineering Wizardry*, Hoboken, NJ: Wiley, 2013.

5. S. Fitzgerald and M. Shiloh, *The Arduino Projects Book*, Torino, Italy: Arduino, 2012.

บทที่ 5 โพรเจกต์ประยุกต์

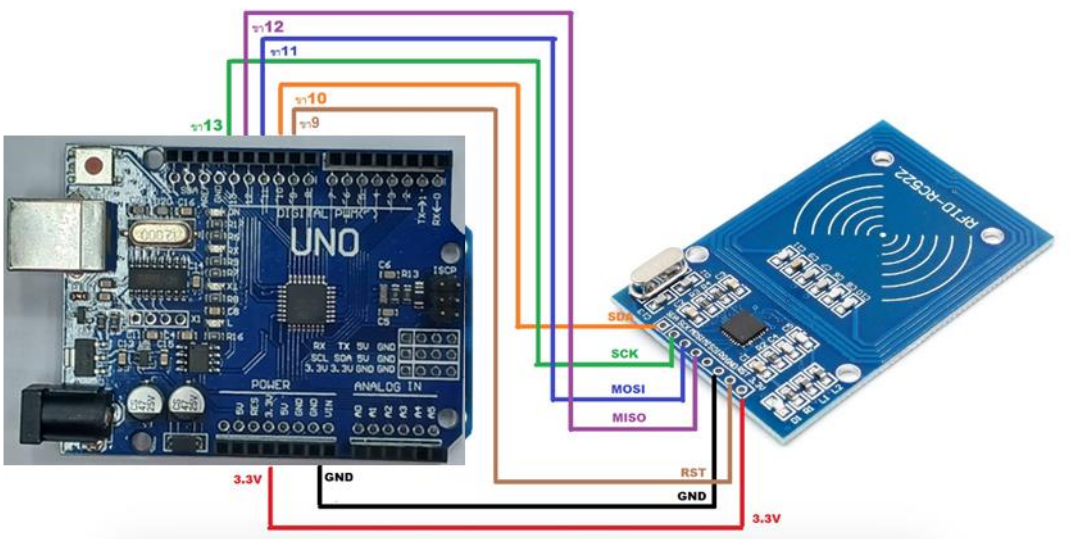
การนำ Arduino มาใช้ในโพรเจกต์อิเล็กทรอนิกส์เป็นการประยุกต์ใช้เทคโนโลยีที่ทันสมัย เพื่อเพิ่มประสิทธิภาพและตอบสนองความต้องการของโพรเจกต์ต่างๆ ในบทนี้ จะนำเสนอวิธีการใช้ Arduino ในการพัฒนาโพรเจกต์อิเล็กทรอนิกส์ ตั้งแต่ขั้นตอนการเริ่มต้นไปจนถึงการติดต่อกับเซ็นเซอร์ การควบคุมอุปกรณ์และการเชื่อมต่อกับระบบภายนอก โปรแกรมตัวอย่างและแนวทางการใช้งานที่นำเสนอในบทนี้จะช่วยให้ผู้อ่านเข้าใจและสามารถประยุกต์ใช้ Arduino ในโพรเจกต์รูปแบบต่าง ๆ ได้

5.1 โพรเจกต์ที่ 1 การใช้งาน Arduino เชื่อมต่อ RFID Module เพื่ออ่านสมาร์ตคาร์ด (Smart Card)

5.1.1 การเชื่อมโยงบอร์ด และอุปกรณ์ต่างๆ

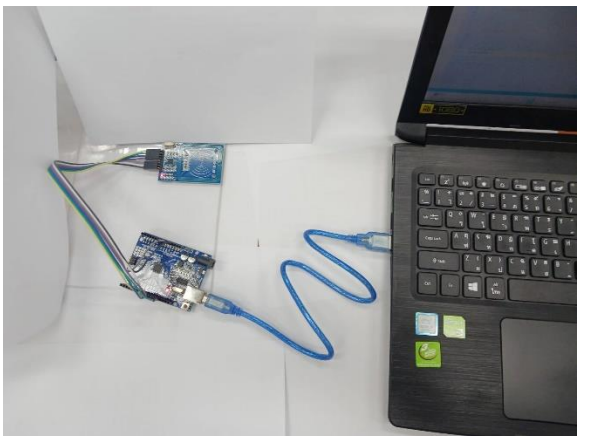
ให้ทำการเชื่อมสายไฟต่อระหว่าง Arduino UNO กับ RFID Module ดังรูปที่ 5.1 ซึ่งจะมีการต่อขาต่างๆ ได้ดังต่อไปนี้

- 3.3V -> Vcc
- GND -> GND
- ขา9 -> RST
- ขา10 -> SDA
- ขา11 -> MOSI
- ขา12 -> MISO
- ขา13 -> SCK



รูปที่ 5.1 การต่อสายไฟ ระหว่าง Arduino UNO กับ RFID Module

เมื่อต่อสายไฟเรียบร้อยแล้ว ให้ทำการต่อสายเชื่อมไปยังคอมพิวเตอร์ต่อไป ดังรูปที่ 5.2

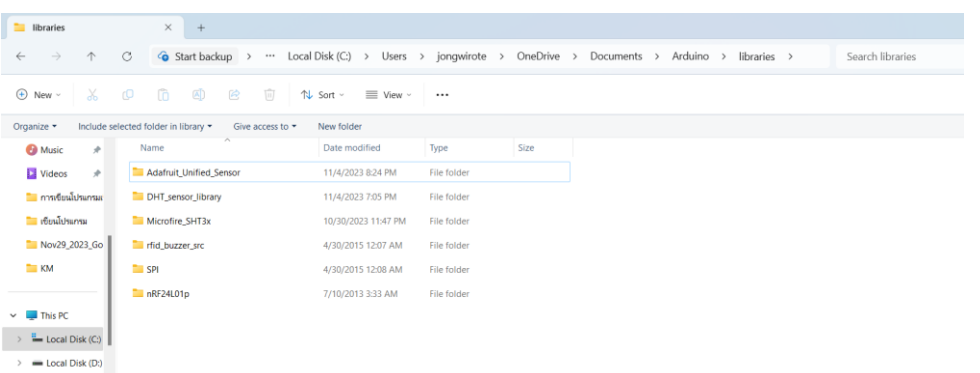


รูปที่ 5.2 การต่อ Arduino UNO, RFID Module และ คอมพิวเตอร์

ให้ทำการโหลด Library RFID Module จากลิงค์ด้านล่าง

http://www.mediafire.com/download/obot7avl7bfuucj/rfid_buzzer.rar

แล้วแตกไฟล์ไปเก็บไว้ที่โฟลเดอร์ Arduino/libraries ดังรูปที่ 5.3 เพื่อให้โปรแกรม Arduino IDE รู้จัก RFID Module ซึ่งแต่ละเครื่องคอมพิวเตอร์จะมีการลงติดตั้งโปรแกรมไม่เหมือนกัน แต่ว่าจะมีชื่อโฟลเดอร์เดียวกัน



รูปที่ 5.3 การจัดเก็บ Libraries

5.1.2 การเขียนโปรแกรม

ให้ทำการเขียนโปรแกรม Arduino IDE ดังนี้

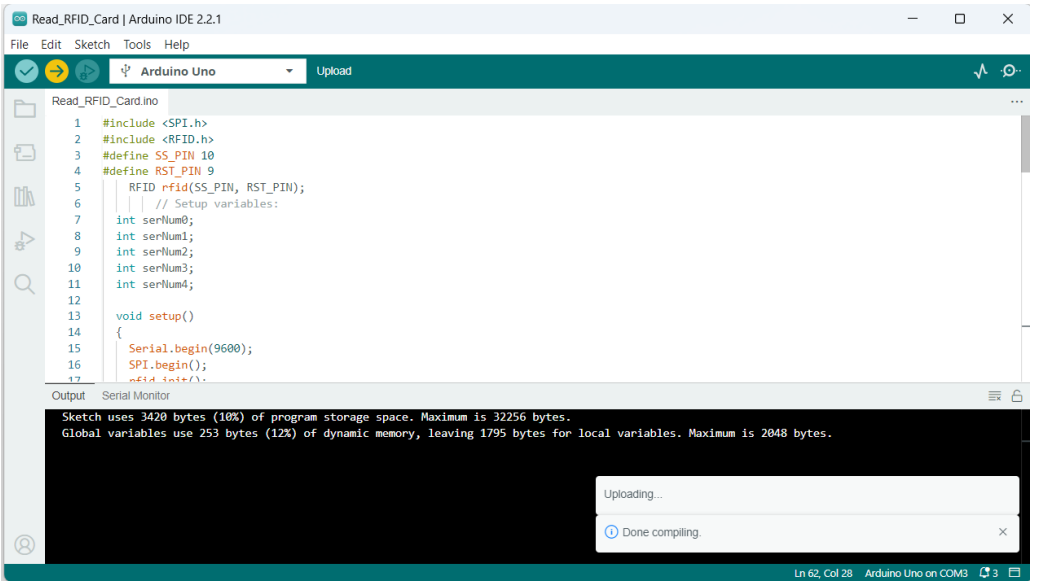
```
#include <SPI.h>// ไลบรารีสำหรับการสื่อสาร SPI
#include <RFID.h>// ไลบรารีสำหรับการสื่อสารกับโมดูล RFID-RC522
#define SS_PIN 10 // กำหนดขา Slave Select (SS) เป็นขา 10 ของ Arduino
#define RST_PIN 9 // กำหนดขา Reset (RST) เป็นขา 9 ของ Arduino
    RFID rfid(SS_PIN, RST_PIN); // สร้างวัตถุ rfid โดยกำหนดขา SS และ RST
```

```
    // ตัวแปรเพื่อเก็บหมายเลข Serial ของสมาร์ตคาร์ด

int serNum0;
int serNum1;
int serNum2;
int serNum3;
int serNum4;
  void setup()
{
  Serial.begin(9600); // เริ่มต้นการสื่อสารแบบอนุกรมที่ความเร็ว 9600 bps
  SPI.begin(); // เริ่มต้นการสื่อสาร SPI
  rfid.init(); // เริ่มต้นการทำงานของโมดูล RFID-RC522
}
  void loop()
{

  if (rfid.isCard()) { // ตรวจสอบว่ามีสมาร์ตคาร์ดอยู่ในระยะของเครื่องอ่านหรือไม่
    if (rfid.readCardSerial()) { // อ่าน Serial Number ของสมาร์ตคาร์ด
    // ตรวจสอบว่า Serial Number ของสมาร์ตคาร์ดที่อ่านได้ไม่ซ้ำกับสมาร์ตคาร์ดที่อ่านก่อนหน้านี้

      if (rfid.serNum[0] != serNum0
          && rfid.serNum[1] != serNum1
          && rfid.serNum[2] != serNum2
          && rfid.serNum[3] != serNum3
          && rfid.serNum[4] != serNum4
         ) {
        // หากเป็นสมาร์ตคาร์ดใบใหม่ แสดง Serial Number ที่อ่านได้
        Serial.println(" ");
        Serial.println("Card found"); // แสดงข้อความว่าเจอสมาร์ตคาร์ดใหม่
        // บันทึก Serial Number ของสมาร์ตคาร์ดในตัวแปร
        serNum0 = rfid.serNum[0];
        serNum1 = rfid.serNum[1];
        serNum2 = rfid.serNum[2];
        serNum3 = rfid.serNum[3];
```

```
        serNum4 = rfid.serNum[4];
        // แสดง Serial Number ในรูปแบบ Decimal
      Serial.println("Cardnumber:");
      Serial.print("Dec: ");
      Serial.print(rfid.serNum[0], DEC);
      Serial.print(", ");
      Serial.print(rfid.serNum[1], DEC);
      Serial.print(", ");
      Serial.print(rfid.serNum[2], DEC);
      Serial.print(", ");
      Serial.print(rfid.serNum[3], DEC);
      Serial.print(", ");
      Serial.print(rfid.serNum[4], DEC);
      Serial.println(" ");
        // แสดง Serial Number ในรูปแบบ Hexadecimal
      Serial.print("Hex: ");
      Serial.print(rfid.serNum[0], HEX);
      Serial.print(", ");
      Serial.print(rfid.serNum[1], HEX);
      Serial.print(", ");
      Serial.print(rfid.serNum[2], HEX);
      Serial.print(", ");
      Serial.print(rfid.serNum[3], HEX);
      Serial.print(", ");
      Serial.print(rfid.serNum[4], HEX);
      Serial.println(" ");
       } else {
      // ถ้าเป็นสมาร์ตคาร์ดใบเดิมที่เจอแล้ว แสดงเครื่องหมายจุด "."
      Serial.print(".");
    }
  }
}
rfid.halt(); // หยุดการทำงานของโมดูล RFID จนกว่าจะเจอสมาร์ตคาร์ดใหม่
```

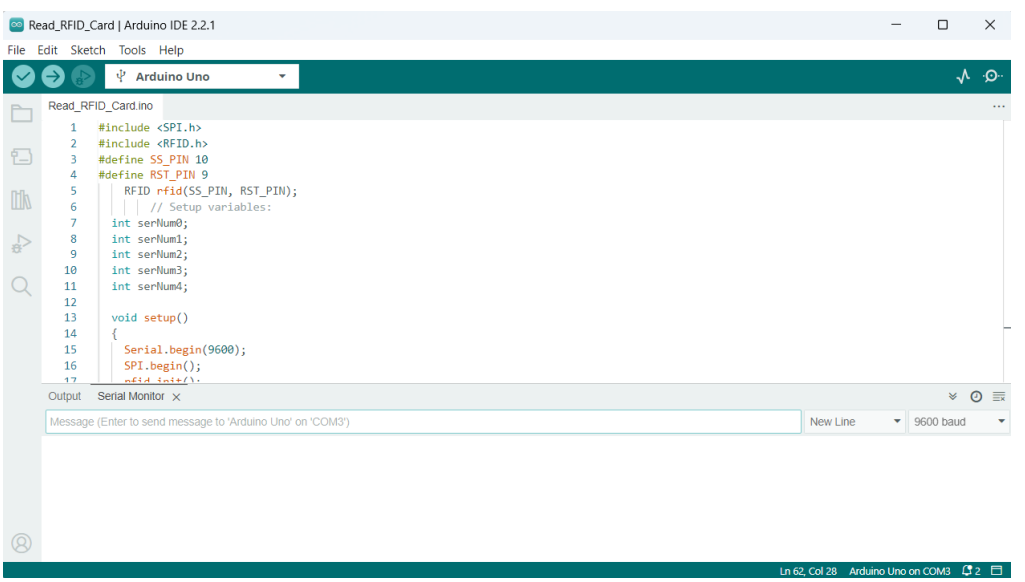
}

แล้วอัปโหลดลงในบอร์ด Arduino จะแสดงการอัปโหลดได้ดังรูปที่ 5.4



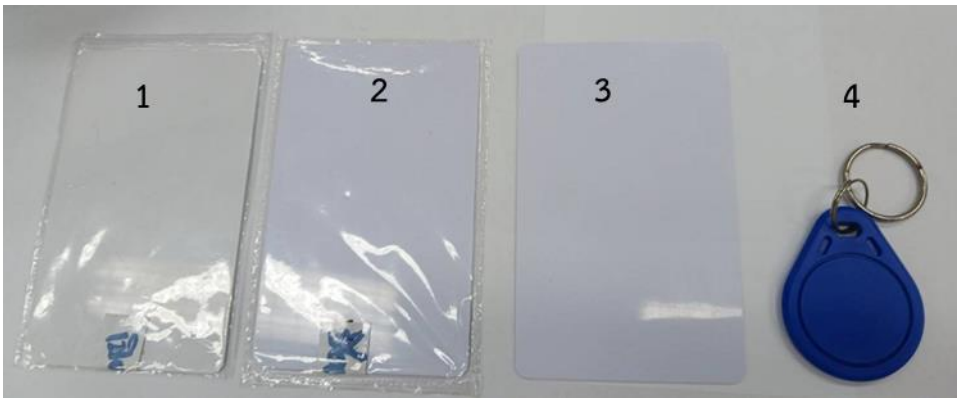
รูปที่ 5.4 การอัปโหลดโปรแกรมลงบอร์ด Arduino

ให้ทำการเปิดเมนู Serial Monitor ดังรูปที่ 5.5 เพื่อทำการอ่านบัตรสมาร์ตคาร์ด



รูปที่ 5.5 การเปิดเมนู Serial Monitor

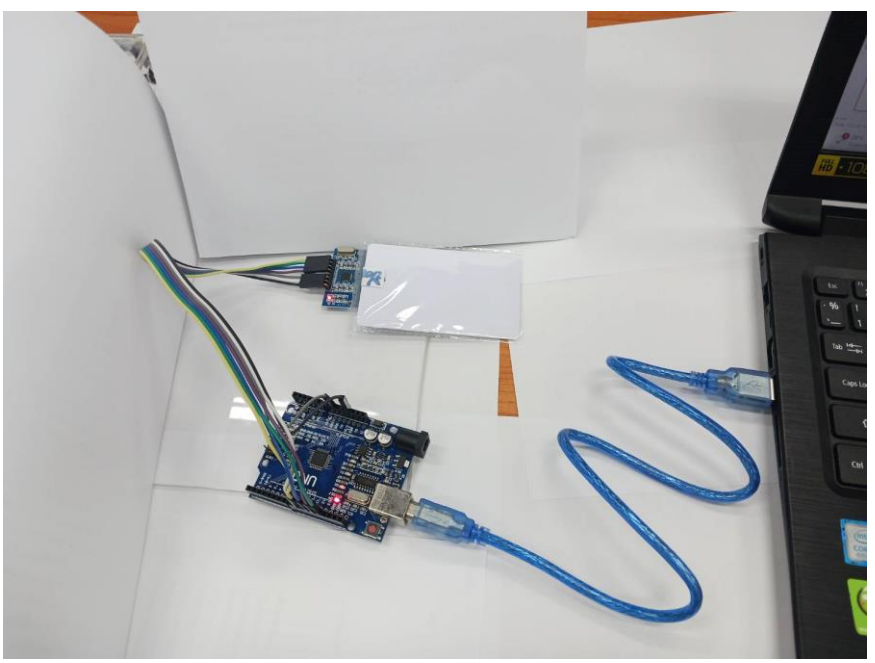
มีบัตรสมาร์ตคาร์ดทั้งหมด 4 ใบแสดงได้ดังรูปที่ 5.6



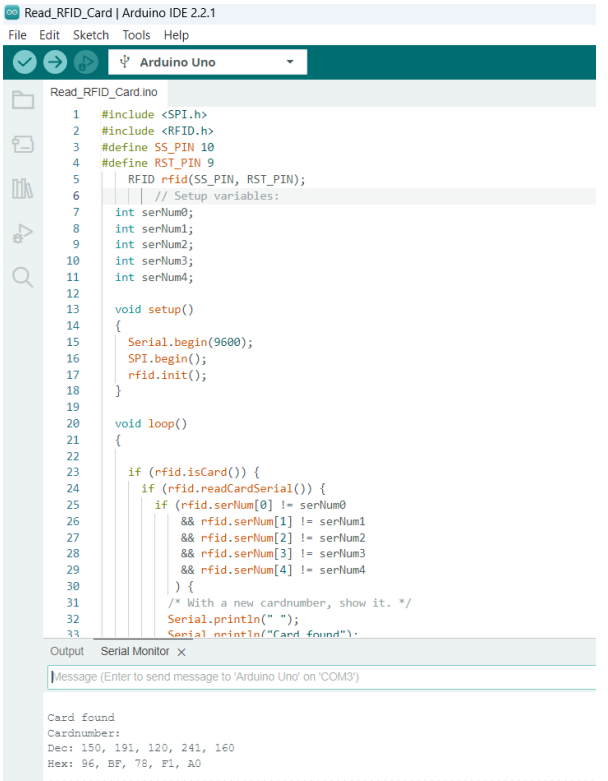
รูปที่ 5.6 บัตรสมาร์ตคาร์ดที่ใช้ทั้ง 4 ใบ

5.1.3 ผลลัพธ์ที่ได้

ให้ทำการแตะบัตรสมาร์ตคาร์ดดังรูปที่ 5.7 เพื่อใช้ในการอ่านค่า ซึ่งจะแสดงผลลัพธ์ได้ดังรูปที่ 5.8



รูปที่ 5.7 การแตะบัตรสมาร์ตคาร์ดเพื่อใช้ในการอ่านค่า



รูปที่ 5.8 ผลลัพธ์ในการอ่านค่าสมาร์ตคาร์ดใน Serial Monitor

5.2 โพรเจกต์ที่ 2 การใช้งาน RFID Card เพื่อใช้ในการเปิด หรือปิดไฟ LED

5.2.1 การเชื่อมโยงบอร์ด และอุปกรณ์ต่างๆ

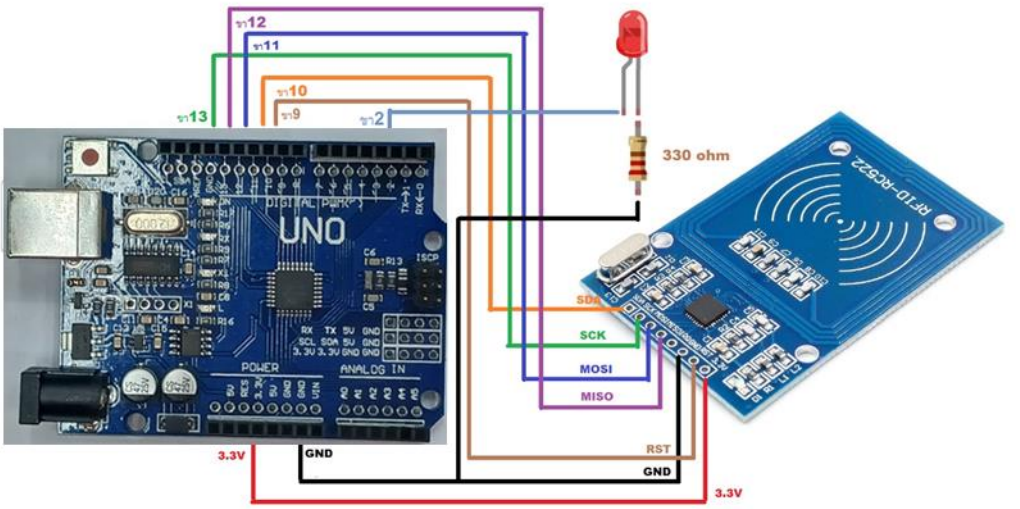
ให้ทำการเชื่อมสายไฟต่อระหว่าง Arduino UNO, RFID Module และหลอดไฟ LED ดังรูปที่ 5.9 ซึ่งจะมีการต่อขาต่างๆ ได้ดังต่อไปนี้

- 3.3V -> Vcc
- GND -> GND
- ขา9 -> RST
- ขา10 -> SDA
- ขา11 -> MOSI
- ขา12 -> MISO
- ขา13 -> SCK

Arduino uno r3 -> LED

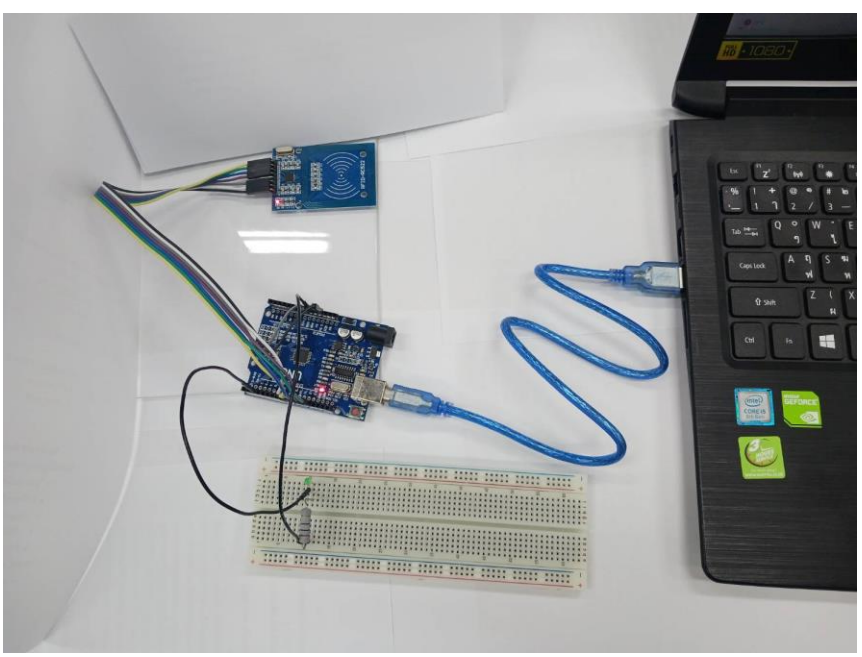
- ขา2 -> LED

- GND -> เข้า LED อีกขา



รูปที่ 5.9 การต่อสายไฟ ระหว่าง Arduino UNO, RFID Module และไฟ LED

ให้ทำการต่ออุปกรณ์ต่างๆ เข้าดังรูปที่ 5.10 จากโพรเจกต์ที่ 1 ทำให้ทราบว่าบัตรสมาร์ตคาร์ดที่อ่านได้มีค่าอะไรบ้าง ก็จะนำค่าดังกล่าวมากำหนดในโปรแกรม Arduino IDE เพื่อเขียนโปรแกรมให้บัตรสมาร์ตคาร์ดที่ต้องการ สามารถเปิดไฟ LED ได้ และถ้าเป็นบัตรสมาร์ตคาร์ดที่ค่าไม่ตรงที่กำหนดในโปรแกรม จะทำให้ไฟดับ



รูปที่ 5.10 การต่อสายไฟ ระหว่าง Arduino UNO, RFID Module,หลอดไฟ LED และคอมพิวเตอร์

5.2.2 การเขียนโปรแกรม

ให้เขียนโปรแกรม ดังนี้

```
#include <SPI.h> // ไลบรารีสำหรับการสื่อสาร SPI
#include <RFID.h> // ไลบรารีสำหรับการทำงานกับโมดูล RFID-RC522
#define SS_PIN 10 // กำหนดขา SS (Slave Select) ของ SPI ให้เป็นขา 10
#define RST_PIN 9 // กำหนดขา RST (Reset) ของโมดูล RFID ให้เป็นขา 9

RFID rfid(SS_PIN, RST_PIN); // สร้างวัตถุ rfid โดยใช้ขา SS และ RST ที่กำหนด

int led1 = 2; // กำหนดขา 2 ของ Arduino สำหรับเชื่อมต่อกับ LED
// ตัวแปรเพื่อเก็บหมายเลข Serial ของสมาร์ตคาร์ด

int serNum0;
int serNum1;
int serNum2;
int serNum3;
int serNum4;
void setup()
{
  Serial.begin(9600); // เริ่มต้นการสื่อสารแบบอนุกรมที่ความเร็ว 9600 bps
  SPI.begin(); // เริ่มต้นการสื่อสาร SPI
  rfid.init(); // เริ่มต้นการทำงานของโมดูล RFID-RC522
  pinMode(led1, OUTPUT); // กำหนดขา 2 (ที่เชื่อมต่อกับ LED) เป็นเอาต์พุต
  digitalWrite(led1, LOW); // ตั้งค่าเริ่มต้นให้ LED ปิด
}
void loop()
{
  if (rfid.isCard()) { // ตรวจสอบว่ามีสมาร์ตคาร์ดอยู่ในระยะของเครื่องอ่านหรือไม่

    if (rfid.readCardSerial()) { // อ่าน Serial Number ของสมาร์ตคาร์ด

// แสดง Serial Number ของสมาร์ตคาร์ดในรูปแบบ Decimal และ Hexadecimal

      Serial.println(" ");
      Serial.println("Card found");
```

```
      serNum0 = rfid.serNum[0];
      serNum1 = rfid.serNum[1];
      serNum2 = rfid.serNum[2];
      serNum3 = rfid.serNum[3];
      serNum4 = rfid.serNum[4];
      //Serial.println(" ");
      Serial.println("Cardnumber:");
      Serial.print("Dec: ");
      Serial.print(rfid.serNum[0], DEC);
      Serial.print(", ");
      Serial.print(rfid.serNum[1], DEC);
      Serial.print(", ");
      Serial.print(rfid.serNum[2], DEC);
      Serial.print(", ");
      Serial.print(rfid.serNum[3], DEC);
      Serial.print(", ");
      Serial.print(rfid.serNum[4], DEC);
      Serial.println(" ");
      Serial.print("Hex: ");
      Serial.print(rfid.serNum[0], HEX);
      Serial.print(", ");
      Serial.print(rfid.serNum[1], HEX);
      Serial.print(", ");
      Serial.print(rfid.serNum[2], HEX);
      Serial.print(", ");
      Serial.print(rfid.serNum[3], HEX);
      Serial.print(", ");
      Serial.print(rfid.serNum[4], HEX);
      Serial.println(" ");
       }
// ตรวจสอบ Serial Number ของสมาร์ตคาร์ดที่อ่านได้และควบคุม LED
    if (rfid.serNum[0] == 150 && rfid.serNum[1] == 191 && rfid.serNum[2] == 120 &&
rfid.serNum[3] == 241 && rfid.serNum[4] == 160) {
```

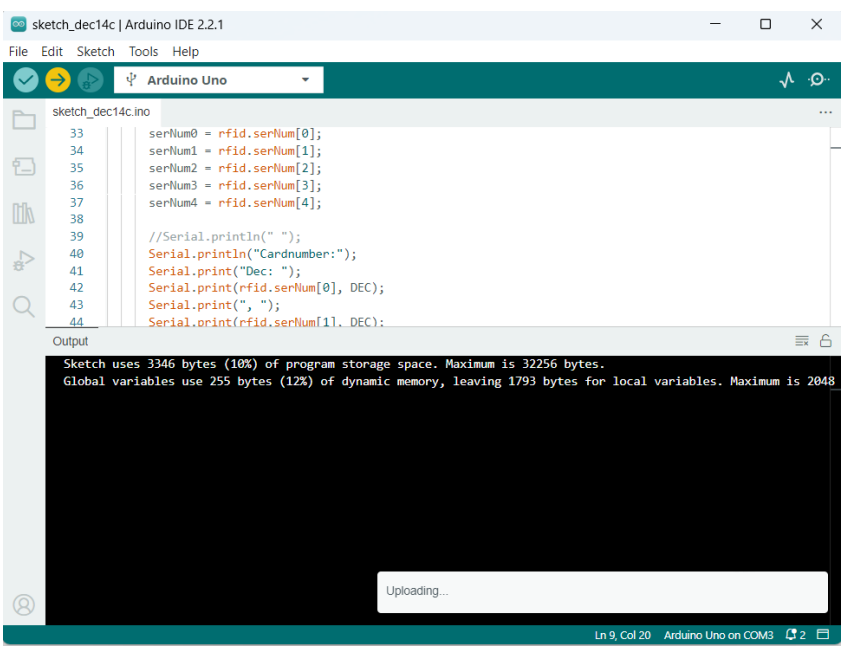
```
// ถ้าหมายเลข Serial ของสมาร์ตคาร์ดตรงกับค่าที่กำหนดไว้ เปิด LED
      digitalWrite(led1, HIGH);
      Serial.println("LED ON");
    }
    else { // ถ้าไม่ตรงกับค่าที่กำหนดไว้ ปิด LED
      digitalWrite(led1, LOW);
      Serial.println("LED OFF");
    }
  }
  rfid.halt(); // หยุดการทำงานของโมดูล RFID จนกว่าจะมีสมาร์ตคาร์ดใหม่เข้ามา
}
```

แล้วเขียนโปรแกรมข้างต้นในโปรแกรม Arduino IDE แล้วทำการอัปโหลดลงบอร์ด Arduino UNO ดังรูปที่ 5.11 โปรแกรมก็พร้อมใช้งาน และจะให้ผลลัพธ์ดังหัวข้อถัดไป

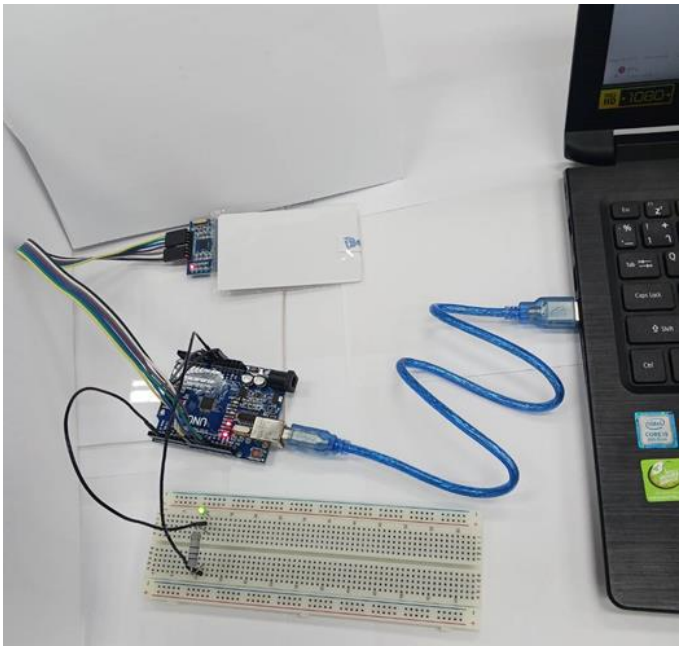
5.2.3 ผลลัพธ์ที่ได้

ในโปรแกรมจะมีกำหนดค่าสมาร์ตคาร์ดที่ใช้ในการเปิดไฟ คือ 150:191:120:241:160

เมื่อนำบัตรสมาร์ตคาร์ดหมายเลข 150:191:120:241:160 มาแตะจะทำให้เกิดไฟ LED สีเขียวสว่าง แสดงได้ดังรูปที่ 5.12

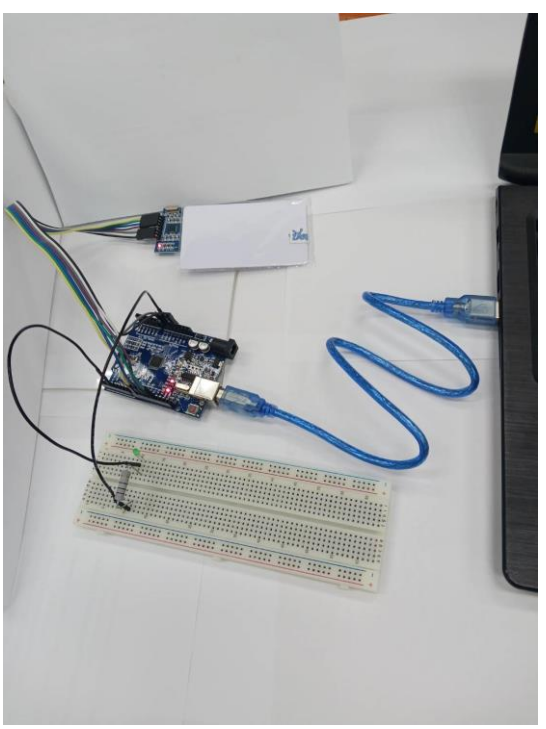


รูปที่ 5.11 การอัปโหลดโปรแกรมลงในบอร์ด Arduino



รูปที่ 5.12 ผลให้เห็นไฟ LED สีเขียวติดสว่าง เมื่อใช้บัตรสมาร์ตคาร์ดที่สามารถเปิดไฟ

และเมื่อนำบัตรสมาร์ตคาร์ดที่หมายเลขอื่นๆ จะทำให้ไฟ LED ดับ แสดงได้ดังรูปที่ 5.13



รูปที่ 5.13 ผลลัพธ์ที่เห็นไฟ LED สีเขียวดับ เมื่อใช้บัตรสมาร์ตคาร์ดอื่นๆ

5.3 โพรเจกต์ที่ 3 การใช้งาน RFID Card เพื่อใช้ในการเปิดไฟ LED ที่ละดวง แล้วปิดไฟ LED ทั้งสองดวง

5.3.1 การเชื่อมโยงบอร์ด และอุปกรณ์ต่างๆ

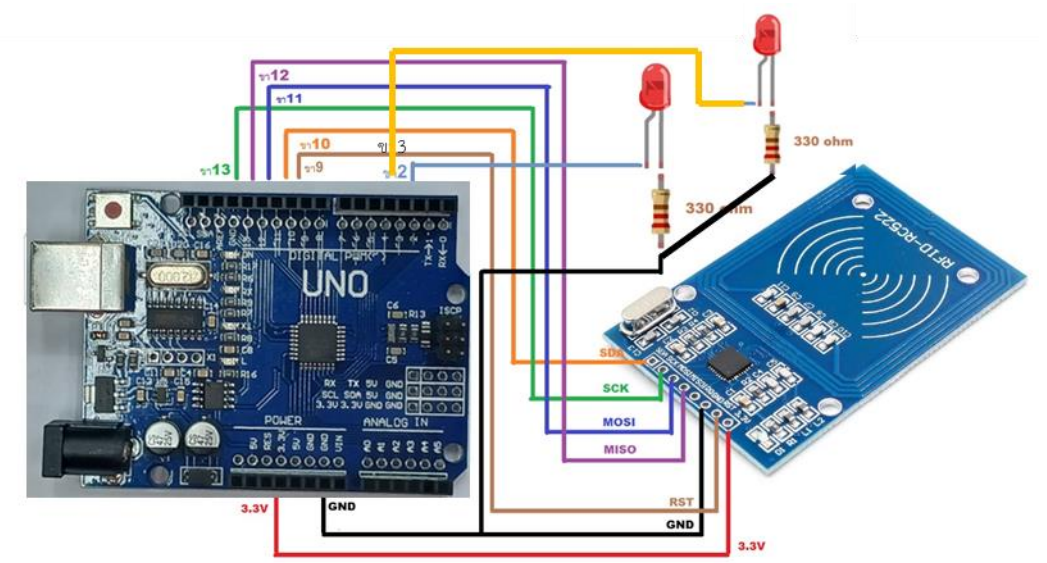
ให้ทำการเชื่อมสายไฟต่อระหว่าง Arduino UNO, RFID Module และหลอดไฟ LED ดังรูปที่ 5.14 ซึ่งจะมีการต่อขาต่างๆ ได้ดังต่อไปนี้

- 3.3V -> Vcc
- GND -> GND
- ขา9 -> RST
- ขา10 -> SDA
- ขา11 -> MOSI
- ขา12 -> MISO
- ขา13 -> SCK

Arduino uno r3 -> LED

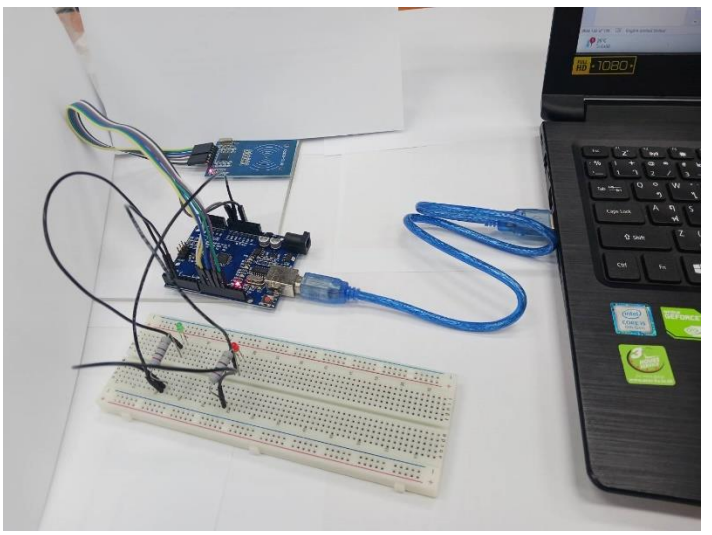
ขา2 -> LED1

- ขา3-> LED2
- GND -> เข้า LED อีกขา ทั้งสองดวง



รูปที่ 5.14 การต่อสายไฟ ระหว่าง Arduino UNO, RFID Module และไฟ LED2 ดวง

ให้ทำการต่ออุปกรณ์ต่างๆ เข้าดังรูปที่ 5.15 จากโพรเจกต์ที่ 1 ทำให้ทราบว่าบัตรสมาร์ตคาร์ดที่อ่านได้มีค่าอะไรบ้าง ก็จะนำค่าดังกล่าวมากำหนดในโปรแกรม Arduino IDE เพื่อเขียนโปรแกรมให้บัตรสมาร์ตคาร์ดที่ต้องการ สามารถเปิดไฟ LED ได้ และถ้าเป็นบัตรสมาร์ตคาร์ดที่ค่าไม่ตรงที่กำหนดในโปรแกรม จะทำให้ไฟดับ



รูปที่ 5.15 การต่อสายไฟ ระหว่าง Arduino UNO, RFID Module,หลอดไฟ LED และคอมพิวเตอร์

5.3.2 การเขียนโปรแกรม

ให้เขียนโปรแกรม ดังนี้

```
#include <SPI.h> // ไลบรารีสำหรับการสื่อสาร SPI
#include <RFID.h> // ไลบรารีสำหรับการทำงานกับโมดูล RFID-RC522
#define SS_PIN 10 // กำหนดขา SS (Slave Select) ของ SPI ให้เป็นขา 10
#define RST_PIN 9 // กำหนดขา RST (Reset) ของโมดูล RFID ให้เป็นขา 9

RFID rfid(SS_PIN, RST_PIN); // สร้างวัตถุ rfid โดยใช้ขา SS และ RST ที่กำหนด
int led1 = 2; // กำหนดขา 2 ของ Arduino สำหรับเชื่อมต่อกับ LED1
int led2 = 3; // กำหนดขา 3 ของ Arduino สำหรับเชื่อมต่อกับ LED2

// ตัวแปรเพื่อเก็บหมายเลข Serial ของสมาร์ตคาร์ด

int serNum0;
int serNum1;
int serNum2;
int serNum3;
```

```
int serNum4;

void setup()
{
  Serial.begin(9600); // เริ่มต้นการสื่อสารแบบอนุกรมที่ความเร็ว 9600 bps
  SPI.begin(); // เริ่มต้นการสื่อสาร SPI
  rfid.init(); // เริ่มต้นการทำงานของโมดูล RFID-RC522
  pinMode(led1, OUTPUT); // กำหนดขา 2 (ที่เชื่อมต่อกับ LED1) เป็นเอาต์พุต
  digitalWrite(led1, LOW); // ตั้งค่าเริ่มต้นให้ LED1 ปิด
  pinMode(led2, OUTPUT); // กำหนดขา 3 (ที่เชื่อมต่อกับ LED2) เป็นเอาต์พุต
  digitalWrite(led2, LOW); // ตั้งค่าเริ่มต้นให้ LED2 ปิด

}
void loop()
{
  if (rfid.isCard()) { // ตรวจสอบว่ามีสมาร์ตคาร์ดอยู่ในระยะของเครื่องอ่านหรือไม่
    if (rfid.readCardSerial()) { // อ่าน Serial Number ของสมาร์ตคาร์ด
      // แสดง Serial Number ของสมาร์ตคาร์ดในรูปแบบ Decimal และ Hexadecimal
      Serial.println(" ");
      Serial.println("Card found");
      serNum0 = rfid.serNum[0]; // เก็บ Serial Number ของสมาร์ตคาร์ดในตัวแปร
      serNum1 = rfid.serNum[1];
      serNum2 = rfid.serNum[2];
      serNum3 = rfid.serNum[3];
      serNum4 = rfid.serNum[4];
      //Serial.println(" ");
      Serial.println("Cardnumber:");
      Serial.print("Dec: ");
      Serial.print(rfid.serNum[0], DEC);
      Serial.print(", ");
      Serial.print(rfid.serNum[1], DEC);
      Serial.print(", ");
      Serial.print(rfid.serNum[2], DEC);
```

```
      Serial.print(", ");
      Serial.print(rfid.serNum[3], DEC);
      Serial.print(", ");
      Serial.print(rfid.serNum[4], DEC);
      Serial.println(" ");

      Serial.print("Hex: ");
      Serial.print(rfid.serNum[0], HEX);
      Serial.print(", ");
      Serial.print(rfid.serNum[1], HEX);
      Serial.print(", ");
      Serial.print(rfid.serNum[2], HEX);
      Serial.print(", ");
      Serial.print(rfid.serNum[3], HEX);
      Serial.print(", ");
      Serial.print(rfid.serNum[4], HEX);
      Serial.println(" ");

    }
// ตรวจสอบ Serial Number ของสมาร์ตคาร์ดที่อ่านได้และควบคุม LED
    if (rfid.serNum[0] == 150 && rfid.serNum[1] == 191 && rfid.serNum[2] == 120 &&
rfid.serNum[3] == 241 && rfid.serNum[4] == 160) {
      // ถ้าหมายเลข Serial ของสมาร์ตคาร์ดตรงกับค่าที่กำหนดไว้ เปิด LED1
     digitalWrite(led1, HIGH);
      Serial.println("LED1 ON");
      delay(1000); // หน่วงเวลาการเปิด LED1 เป็นเวลา 1 วินาที
    }
    if (rfid.serNum[0] == 14 && rfid.serNum[1] == 208 && rfid.serNum[2] == 219 &&
rfid.serNum[3] == 115 && rfid.serNum[4] == 118) {
    // ถ้าหมายเลข Serial ของสมาร์ตคาร์ดตรงกับค่าที่กำหนดไว้ เปิด LED2
      digitalWrite(led2, HIGH);
      Serial.println("LED2 ON");
    }
```

```
    if (rfid.serNum[0] == 134 && rfid.serNum[1] == 13 && rfid.serNum[2] == 179 &&
rfid.serNum[3] == 31 && rfid.serNum[4] == 39) {
// ถ้าหมายเลข Serial ของสมาร์ตคาร์ดตรงกับค่าที่กำหนดไว้ ปิดทั้ง LED1 และ LED2
      digitalWrite(led1, LOW);
      Serial.println("LED1 OFF");
      digitalWrite(led2, LOW);
      Serial.println("LED2 OFF");
    }
  }
  rfid.halt(); // หยุดการทำงานของโมดูล RFID จนกว่าจะมีสมาร์ตคาร์ดใหม่เข้ามา
}
```

ให้ทำการเขียนโปรแกรมข้างต้น ลงในโปรแกรม Arduino IDE แล้วให้ทำการอัปโหลดโปรแกรมลงบอร์ด Arduino UNO เมื่ออัปโหลดโปรแกรมเรียบร้อย ก็จะสามารถให้ผลลัพธ์การทดสอบได้ดังนี้

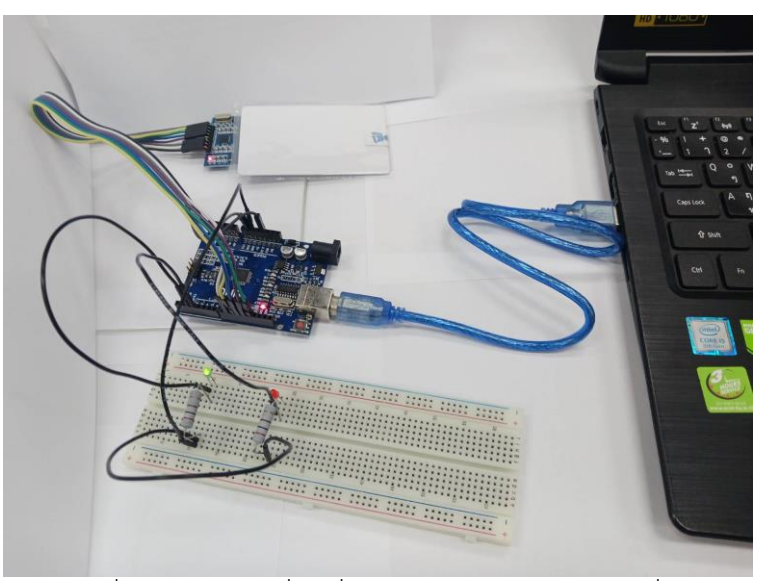
5.3.3 ผลลัพธ์ที่ได้

ค่าสมาร์ตคาร์ดที่ใช้ในการเปิดไฟดวงที่ 1 คือ 150:191:120:241:160

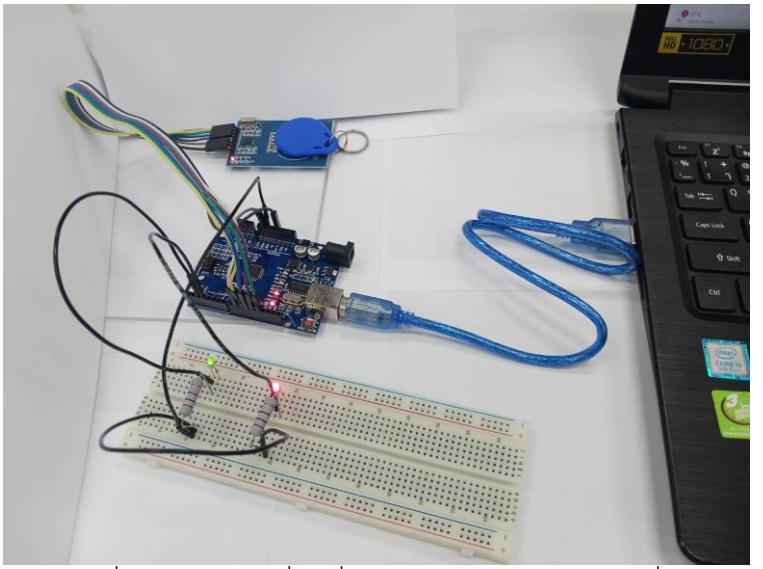
ค่าสมาร์ตคาร์ดที่ใช้ในการเปิดไฟดวงที่ 2 คือ 14:208:219:115:118

และจะใช้ค่าสมาร์ตคาร์ดที่ใช้ในการปิดไฟทั้งสองดวง พร้อมกัน คือ 134:13:179:31:39

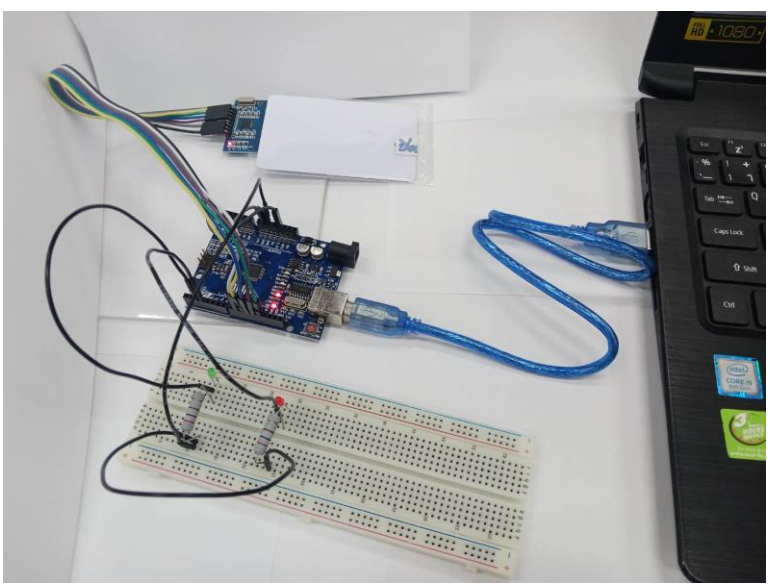
ซึ่งการนำสมาร์ตคาร์ดที่มีค่าที่กำหนดในการเปิดไฟดวงที่ 1 และดวงที่ 2 มาแตะจะทำให้เกิดผลลัพธ์แสดงได้ดังรูปที่ 5.16 และ 5.17 แล้วถ้านำสมาร์ตคาร์ดที่ใช้สำหรับดับไฟทั้งสองดวง มาแตะจะให้เกิดผลลัพธ์ไฟดับทั้งสองดวงดังรูปที่ 5.18 โดยที่จะสามารถเห็นผลลัพธ์ที่ Serial Monitor ได้ดังรูปที่ 5.19-5.21 ตามลำดับ



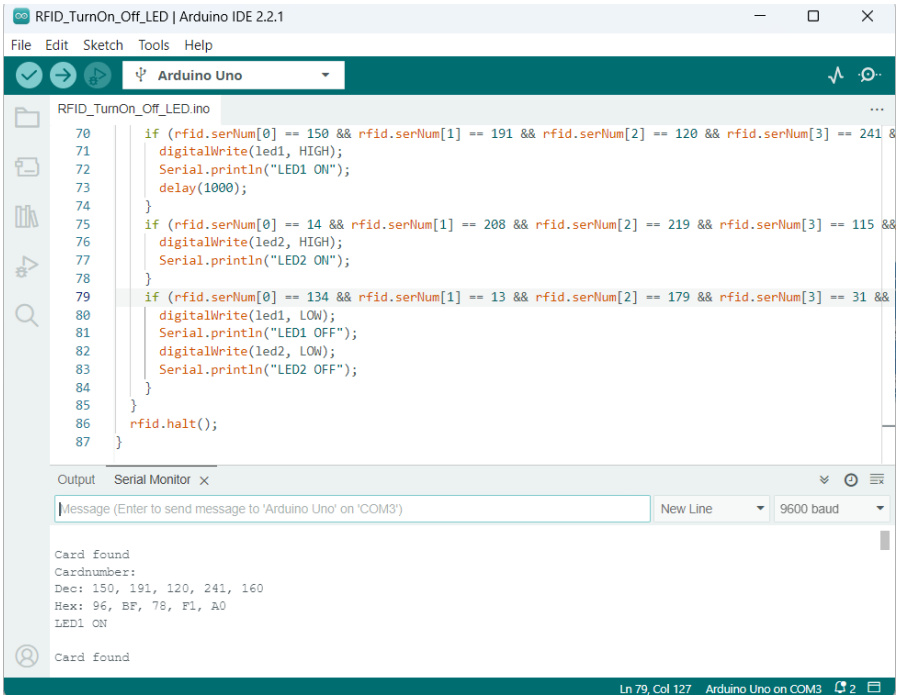
รูปที่ 5.16 ผลลัพธ์ที่ได้เมื่อใช้สมาร์ตคาร์ดเปิดไฟดวงที่ 1



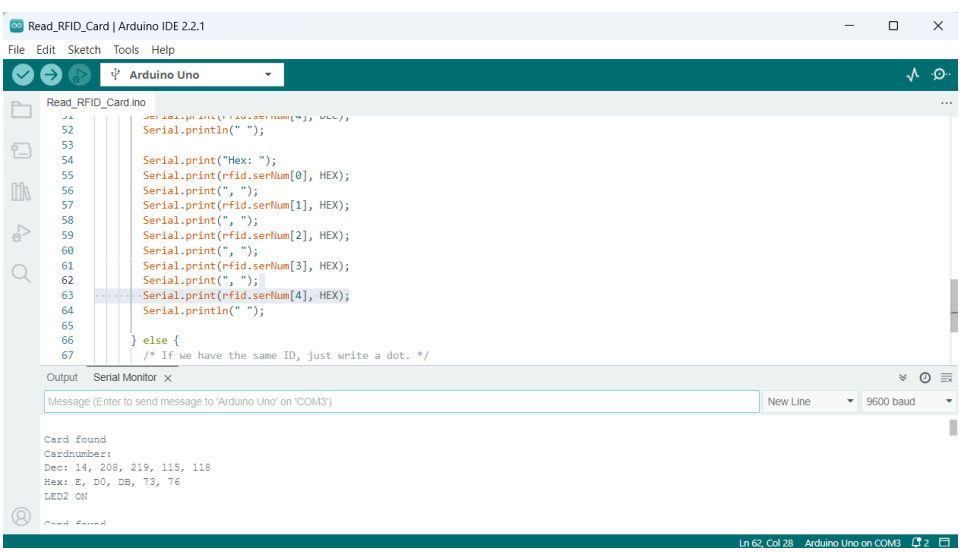
รูปที่ 5.17 ผลลัพธ์ที่ได้เมื่อใช้สมาร์ตคาร์ดเปิดไฟดวงที่ 2



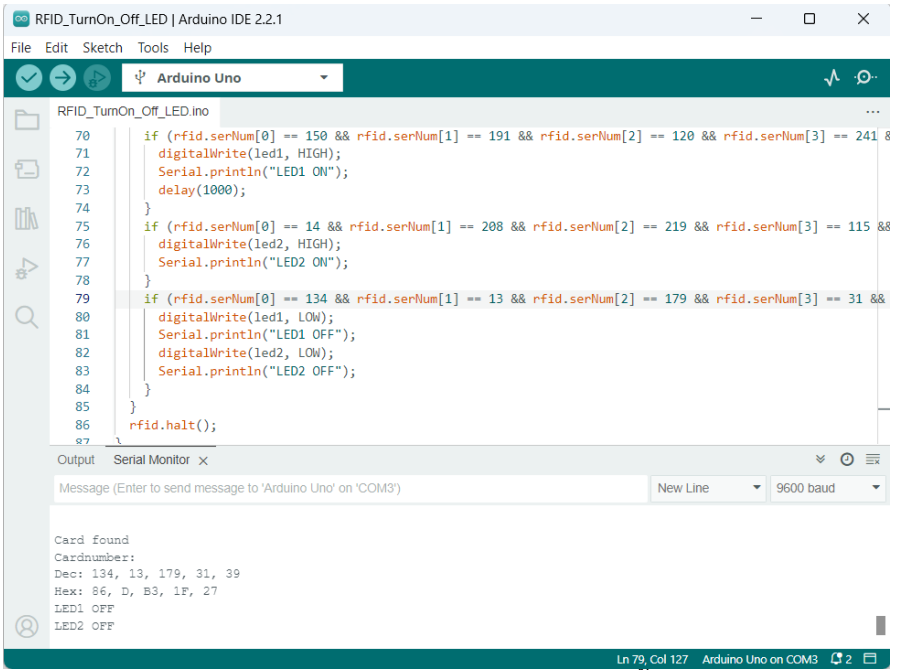
รูปที่ 5.18 ผลลัพธ์ที่ได้เมื่อใช้สมาร์ตคาร์ดปิดไฟทั้งสองดวง



รูปที่ 5.19 ผลลัพธ์ที่ Serial Monitor เปิดไฟดวงที่ 1



รูปที่ 5.20 ผลลัพธ์ที่ Serial Monitor เปิดไฟดวงที่ 2



รูปที่ 5.21 ผลลัพธ์ที่ Serial Monitor ปิดไฟทั้งสองดวง

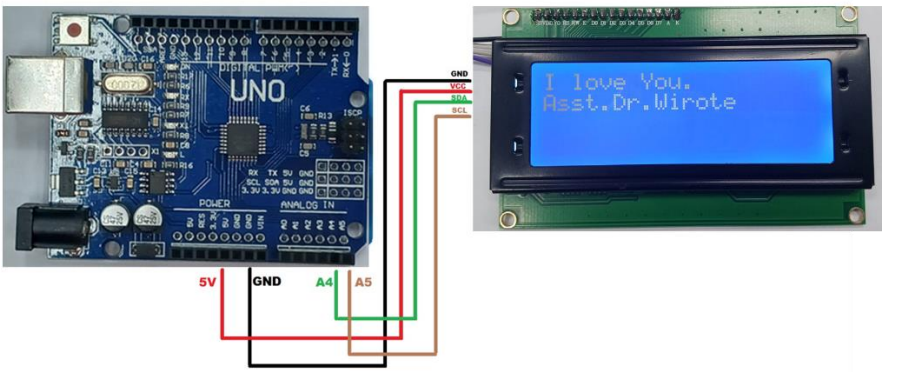
5.4 โพรเจกต์ที่ 4 การใช้งาน Arduino ในการวัดความชื้นและอุณหภูมิ พร้อมแสดงผลผ่านจอ LCD 1602 แบบ I2C

5.4.1 การเชื่อมโยงบอร์ด และอุปกรณ์ต่างๆ

วิธีการต่ออุปกรณ์ จอแสดงผล LCD 1602 -> Arduino uno r3 ให้ทำการเชื่อมสาย และขาของบอร์ด Arduino กับ ขาของ LCD1602 ดังนี้

- gnd -> GND
- Vcc -> 5V
- SDA -> ขา A4 (สำหรับ Arduino uno r3)
- SCL -> ขา A5 (สำหรับ Arduino uno r3)

แสดงการเชื่อมต่อได้ดังรูปที่ 5.22

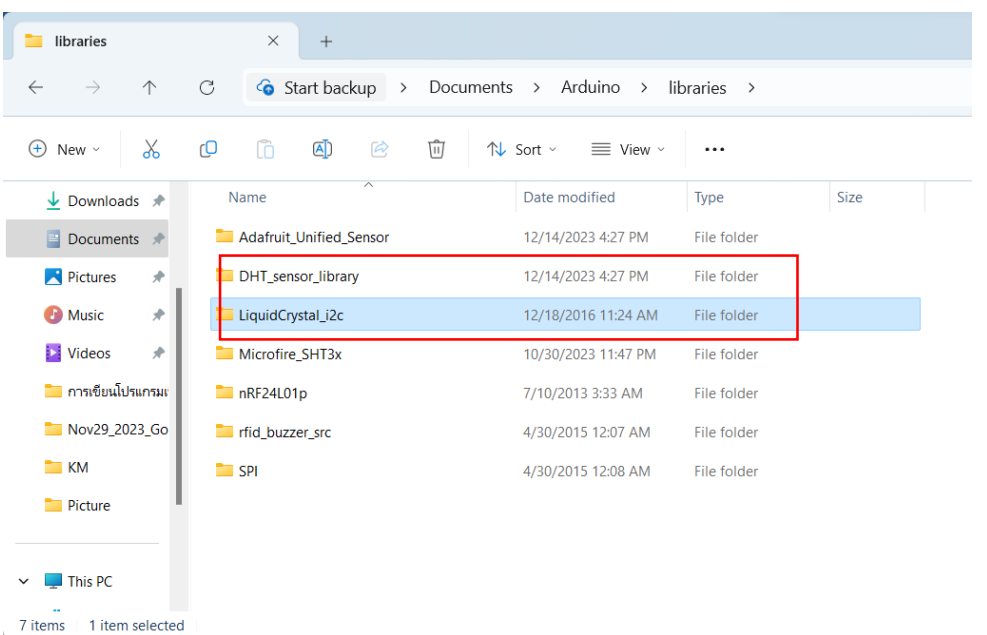


รูปที่ 5.22 การเชื่อมต่อระหว่างขาบอร์ด Arduino กับ LCD 1602

ให้ทำการโหลด Library LCD1602 จากลิงค์ด้านล่าง

- http://www.mediafire.com/file/iavimlofgo1aq6l/LiquidCrystal_i2c.rar
- http://download.ab.in.th/download.php?file=DHT-sensor-library-master.zip

แล้วแตกไฟล์ไปเก็บไว้ที่โฟลเดอร์ Arduino/libraries ดังรูปที่ 5.23 เพื่อให้โปรแกรม Arduino IDE รู้จัก LCD 1602 และเซ็นเซอร์ DHT22 ซึ่งแต่ละเครื่องคอมพิวเตอร์จะมีการลงติดตั้งโปรแกรมไม่เหมือนกัน แต่ว่าจะมีชื่อโฟลเดอร์เดียวกัน



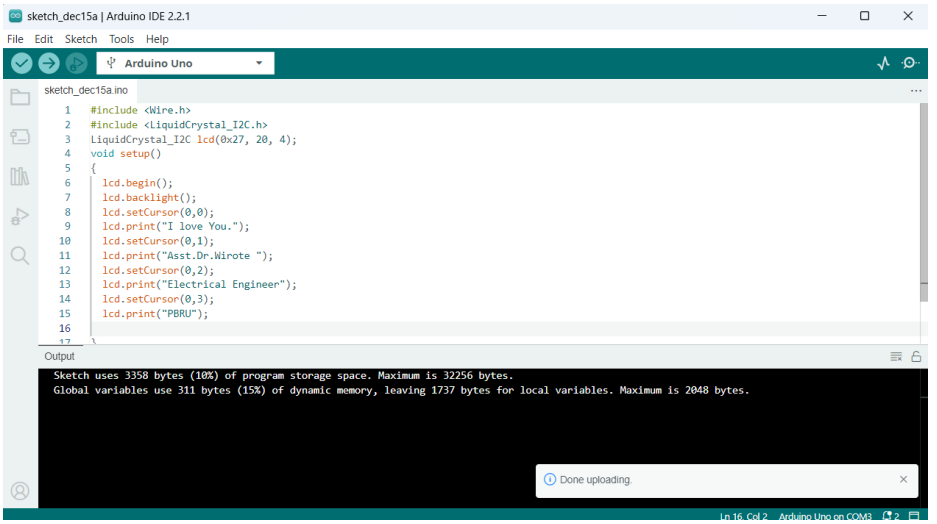
รูปที่ 5.23 โฟลเดอร์ในการจัดเก็บ Libraries ของ LCD 1602

5.4.2 การเขียนโปรแกรม

ให้ทำการเขียนโปรแกรม ดังนี้

```
#include <Wire.h> // ไลบรารีสำหรับการสื่อสาร I2C
#include <LiquidCrystal_I2C.h> // ไลบรารีสำหรับการควบคุมจอ LCD ผ่าน I2C
// สร้างวัตถุ LiquidCrystal_I2C สำหรับควบคุมจอ LCD ที่มีที่อยู่ I2C 0x27 และขนาด 20x4

LiquidCrystal_I2C lcd(0x27, 20, 4);
void setup()
{
  lcd.begin(); // เริ่มต้นการสื่อสารกับจอ LCD
  lcd.backlight(); // เปิดไฟพื้นหลังของจอ LCD
// กำหนดตำแหน่งของเคอร์เซอร์ที่บรรทัดที่ 1 (เริ่มจากตำแหน่ง 0) และคอลัมน์ที่ 1 (เริ่มจากตำแหน่ง 0)
  lcd.setCursor(0,0);
  lcd.print("I love You."); // พิมพ์ข้อความ "I love You." ที่บรรทัดที่ 1
  lcd.setCursor(0,1); // กำหนดตำแหน่งของเคอร์เซอร์ที่บรรทัดที่ 2 และคอลัมน์ที่ 1
  lcd.print("Asst.Dr.Wirote "); // พิมพ์ข้อความ "Asst.Dr.Wirote " ที่บรรทัดที่ 2
  lcd.setCursor(0,2); // กำหนดตำแหน่งของเคอร์เซอร์ที่บรรทัดที่ 3 และคอลัมน์ที่ 1
  lcd.print("Electrical Engineer"); // พิมพ์ข้อความ "Electrical Engineer" ที่บรรทัดที่ 3
  lcd.setCursor(0,3); // กำหนดตำแหน่งของเคอร์เซอร์ที่บรรทัดที่ 4 และคอลัมน์ที่ 1
  lcd.print("PBRU"); // พิมพ์ข้อความ "PBRU" ที่บรรทัดที่ 4
 }
void loop() { // ไม่มีการกระทำใด ๆ ใน loop นี้ โค้ดทั้งหมดอยู่ในฟังก์ชัน setup
}
```

แล้วให้ทำการอัปโหลดเข้าบอร์ด Arduino แสดงได้ดังรูปที่ 5.24 ก็จะทำให้บอร์ด Arduino UNO สามารถทำงานกับบัตรสมาร์ตคาร์ดได้ตามที่กำหนดในโปรแกรมข้างต้น

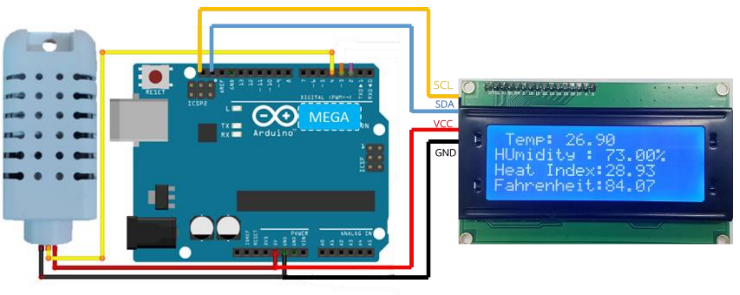


รูปที่ 5.24 การอัปโหลดโปรแกรมลงบอร์ด Arduino

5.4.3 ผลลัพธ์ที่ได้

ระบบจะแสดงผลลัพธ์ได้ดังรูปที่ 5.25 การต่อเซ็นเซอร์ DHT22 และ LCD1602 เข้ากับบอร์ด Arduino Mega2560 จึงมีการเปลี่ยนแปลงขาจากบอร์ด Arduino UNO กับ LCD1602 ในส่วนที่เป็นขา SDA และ SCL ซึ่งจะต้องต่อดังรูปที่ 5.26

รูปที่ 5.25 ผลลัพธ์ในจอ LCD



รูปที่ 5.26 การต่อเซ็นเซอร์ และ LCD กับบอร์ด Arduino mega2560

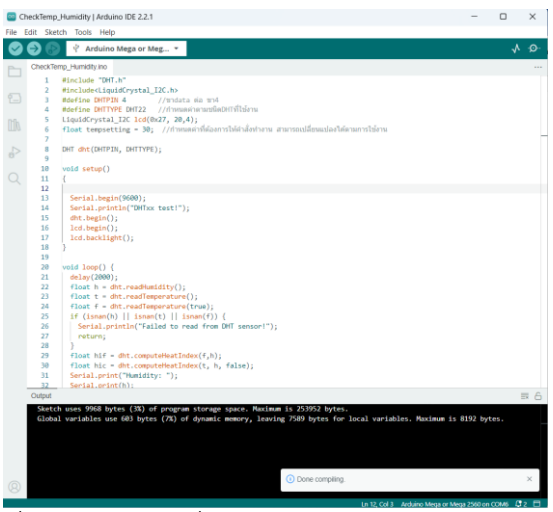
ให้เขียนโปรแกรมดังนี้

```
#include "DHT.h" // นำเข้าไลบรารีสำหรับเซ็นเซอร์ DHT
#include<LiquidCrystal_I2C.h> // นำเข้าไลบรารีสำหรับควบคุมจอ LCD ผ่าน I2C
#define DHTPIN 4         // ขา data ของเซ็นเซอร์ DHT22 เชื่อมต่อกับขา 4 ของ Arduino
#define DHTTYPE DHT22    // กำหนดชนิดของเซ็นเซอร์เป็น DHT22
// สร้างวัตถุ LiquidCrystal_I2C สำหรับจอ LCD ที่มีที่อยู่ I2C 0x27 และขนาด 20x4

LiquidCrystal_I2C lcd(0x27, 20,4);

DHT dht(DHTPIN, DHTTYPE); /* สร้างวัตถุ DHT สำหรับการใช้งานเซ็นเซอร์ DHT ที่ขา DHTPIN และชนิด
DHTTYPE*/

void setup()
{

  Serial.begin(9600); // เริ่มต้นการสื่อสารผ่าน Serial ที่ความเร็ว 9600 bps
  Serial.println("DHTxx test!"); // พิมพ์ข้อความเริ่มต้นสำหรับการทดสอบเซ็นเซอร์ DHT

  dht.begin(); // เริ่มต้นการทำงานของเซ็นเซอร์ DHT
  lcd.begin(); // เริ่มต้นการทำงานของจอ LCD
  lcd.backlight(); // เปิดไฟพื้นหลังของจอ LCD
}

void loop() {
  delay(2000); // หน่วงเวลา 2 วินาที (2000 มิลลิวินาที) ระหว่างการอ่านค่าเซ็นเซอร์แต่ละครั้ง
```

```
  float h = dht.readHumidity(); // อ่านค่าความชื้นจากเซ็นเซอร์ DHT
  float t = dht.readTemperature(); // อ่านค่าอุณหภูมิในหน่วยเซลเซียสจากเซ็นเซอร์ DHT
  float f = dht.readTemperature(true); // อ่านค่าอุณหภูมิในหน่วยฟาเรนไฮต์จากเซ็นเซอร์ DHT
  if (isnan(h) || isnan(t) || isnan(f)) { // ตรวจสอบว่าค่าที่อ่านได้ไม่เป็น NaN (ไม่ใช่ตัวเลข)
    Serial.println("Failed to read from DHT sensor!"); /* ถ้าเป็น NaN ให้พิมพ์ข้อความแสดง
ข้อผิดพลาด*/
    return; // ออกจากฟังก์ชัน loop
  }
  float hif = dht.computeHeatIndex(f,h); /* คำนวณค่า Heat Index (ดัชนีความร้อน) จากอุณหภูมิและ
ความชื้นในหน่วยฟาเรนไฮต์*/

  float hic = dht.computeHeatIndex(t, h, false); /* คำนวณค่า Heat Index จากอุณหภูมิและความชื้น
ในหน่วยเซลเซียส*/


  Serial.print("Humidity: "); //พิมพ์ข้อความ "Humidity: "
  Serial.print(h); //พิมพ์ค่าความชื้น
  Serial.print(" %\t"); //พิมพ์สัญลักษณ์ % และแท็บ
  Serial.print("Temperature: "); //พิมพ์ข้อความ "Temperature: "
  Serial.print(t); //พิมพ์ค่าอุณหภูมิในหน่วยเซลเซียส
  Serial.print(" *C\t"); //พิมพ์สัญลักษณ์ *C และแท็บ
  Serial.print("Heat Index:"); //พิมพ์ข้อความ "Heat Index: "
  Serial.print(hic); //พิมพ์ค่า Heat Index ในหน่วยเซลเซียส
  Serial.print("\t"); //พิมพ์แท็บ
  Serial.print(f); //พิมพ์ค่าอุณหภูมิในหน่วยฟาเรนไฮต์
  Serial.println(" *F\t"); //พิมพ์สัญลักษณ์ *F และขึ้นบรรทัดใหม่
  lcd.home(); // ย้ายเคอร์เซอร์ไปยังตำแหน่งเริ่มต้นของจอ LCD (0,0)
  lcd.print(" Temp: "); // พิมพ์ข้อความ " Temp: "

  lcd.print(t); // พิมพ์ค่าอุณหภูมิในหน่วยเซลเซียส
  lcd.print(" *C"); // พิมพ์สัญลักษณ์ `*C`
  lcd.setCursor(0,1); // ย้ายเคอร์เซอร์ไปยังตำแหน่งเริ่มต้นของบรรทัดที่ 2 (0,1)
  lcd.print("HUmidity : "); // พิมพ์ข้อความ "Humidity :
  lcd.print(h); // พิมพ์ค่าความชื้น
```

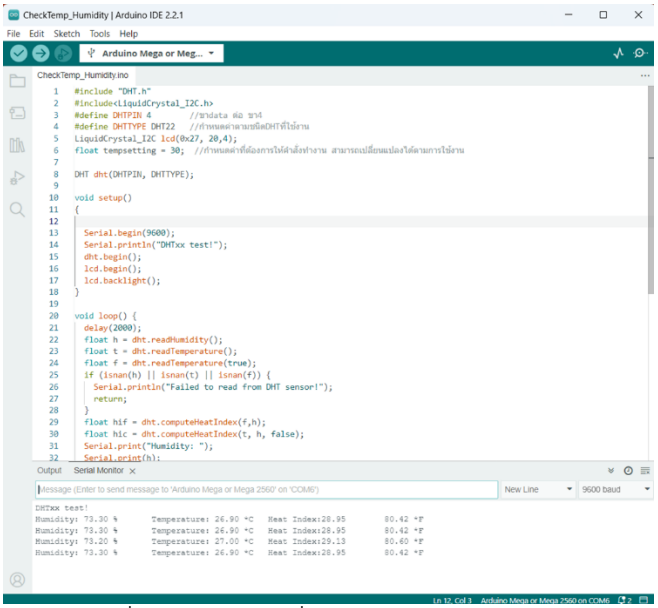
```
  lcd.print("%"); // พิมพ์สัญลักษณ์ `%`
  lcd.setCursor(0,2); // ย้ายเคอร์เซอร์ไปยังตำแหน่งเริ่มต้นของบรรทัดที่ 3 (0,1)
  lcd.print("Heat Index:"); // พิมพ์ข้อความ " Heat Index: "

  lcd.print(hic); // พิมพ์ค่าHeat Index
  lcd.setCursor(0,3); // ย้ายเคอร์เซอร์ไปยังตำแหน่งเริ่มต้นของบรรทัดที่ 4 (0,1)
  lcd.print("Fahrenheit:"); // พิมพ์ข้อความ " Fahrenheit: "
  lcd.print(f); // พิมพ์ค่า Fahrenheit
  lcd.print(" *F"); // พิมพ์สัญลักษณ์ `*F`
}
```

เมื่อเขียนโปรแกรมข้างต้นเสร็จ แล้วทำการกดเครื่องหมายถูกเพื่อให้โปรแกรมคอมไพล์ แสดงได้ดังรูปที่ 5.27



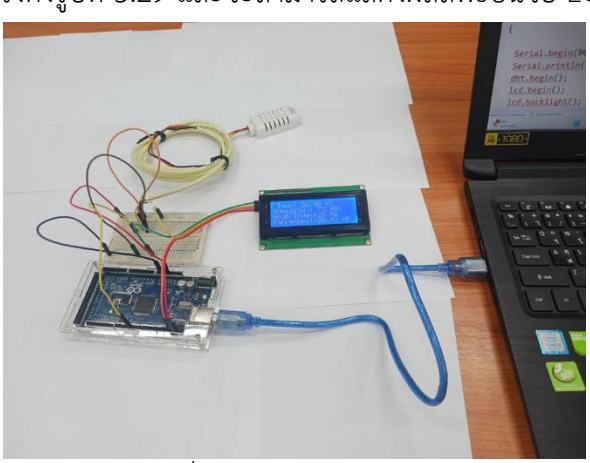
รูปที่ 5.27 ผลลัพธ์เมื่อโปรแกรมคอมไพล์เสร็จเรียบร้อย

ให้ทำการเลือก Serial Monitor เพื่อที่จะได้เห็นผลลัพธ์ดังรูปที่ 5.28



รูปที่ 5.28 ผลลัพธ์ที่ได้ จาก Serial Monitor

แสดงการต่อวงจรจริงดังรูปที่ 5.29 และจะสามารถแสดงผลลัพธ์บนจอ LCD ได้ดังรูปที่ 5.30



รูปที่ 5.29 การต่อวงจรจริง

รูปที่ 5.30 ผลลัพธ์ที่ได้ในจอ LCD

5.5 โพรเจกต์ที่ 5 การใช้งาน Arduino ในการนับคนเข้าออกอัตโนมัติ

5.5.1 การเชื่อมโยงบอร์ด และอุปกรณ์ต่างๆ

ให้ทำการต่อวงจรที่แสดงได้ดังรูปที่ 5.31 แล้วจะเห็นการแสดงการต่อุปกรณ์ LCD, เซ็นเซอร์อินฟราเรด บนบอร์ด Arduino UNO ที่เกิดขึ้นจริงได้ดังรูปที่ 5.32

5.5.2 การเขียนโปรแกรม

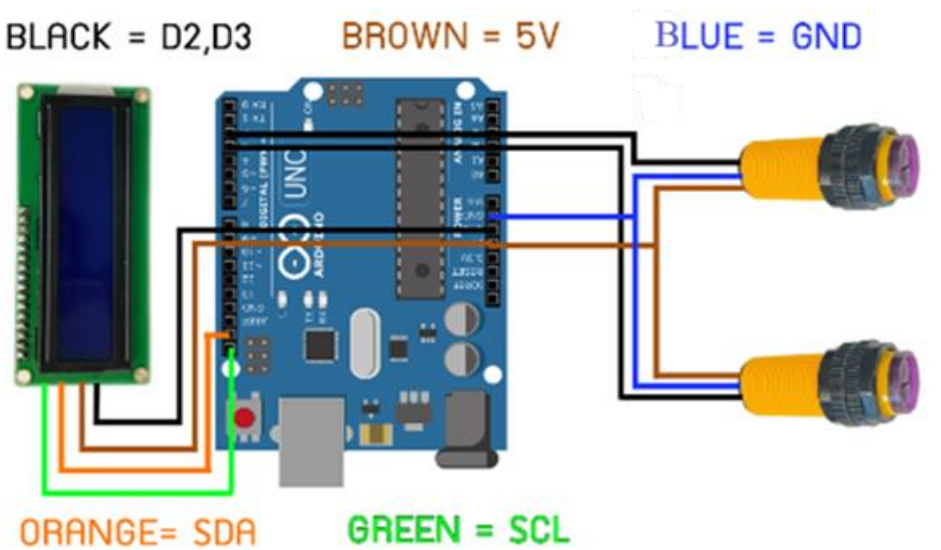
ให้ทำเขียนโปรแกรมดังนี้

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
/* สร้างวัตถุ LiquidCrystal_I2C โดยกำหนดที่อยู่ I2C เป็น 0x27 และขนาดจอ LCD เป็น 16 ตัวอักษรต่อแนวนอน 4 บรรทัด*/
LiquidCrystal_I2C lcd(0x27, 16, 4);
#define IR_IN 2 // กำหนดขาของเซ็นเซอร์อินฟราเรด (IR sensor) ที่ใช้ตรวจจับการเข้าเป็นขาที่ 2

#define IR_OUT 3 // กำหนดขาของเซ็นเซอร์อินฟราเรด (IR sensor) ที่ใช้ตรวจจับการออกเป็นขาที่ 3
int last_In =  -1; // ประกาศตัวแปรเก็บสถานะของขา IR_IN โดยกำหนดค่าเริ่มต้นเป็น -1
int last_Out =  -1; // ประกาศตัวแปรเก็บสถานะของขา IR_OUT โดยกำหนดค่าเริ่มต้นเป็น -1
long walkOutTime; // ประกาศตัวแปรเก็บเวลาเมื่อมีการออก
long walkInTime; // ประกาศตัวแปรเก็บเวลาเมื่อมีการเข้า
int counter_in = 0; // ประกาศตัวแปรนับจำนวนคนที่เข้า
int counter_out = 0; // ประกาศตัวแปรนับจำนวนคนที่ออก
bool isWalkIn = false; // ประกาศตัวแปรบอกสถานะการเข้า
bool isWalkOut = false; // ประกาศตัวแปรบอกสถานะการออก
void setup() {
  Serial.begin(9600); // เริ่มต้นการสื่อสารผ่าน Serial communication ที่อัตราส่งข้อมูล 9600 bps
  lcd.begin(); // เริ่มต้นการใช้งานจอ LCD
  lcd.backlight(); // เปิดไฟหลังจอ LCD
  lcd.setCursor(0, 0); // ตั้งค่าตำแหน่ง cursor ที่แถวที่ 0 และคอลัมน์ที่ 0
  lcd.print("IN  = "); // แสดงข้อความ "IN  = " บนจอ LCD
  lcd.setCursor(0, 1); // ตั้งค่าตำแหน่ง cursor ที่แถวที่ 1 และคอลัมน์ที่ 0

  lcd.print("OUT = "); // แสดงข้อความ "OUT = " บนจอ LCD
  pinMode(IR_IN, INPUT); // กำหนดขาของเซ็นเซอร์อินฟราเรดเป็นขาอินพุต
```

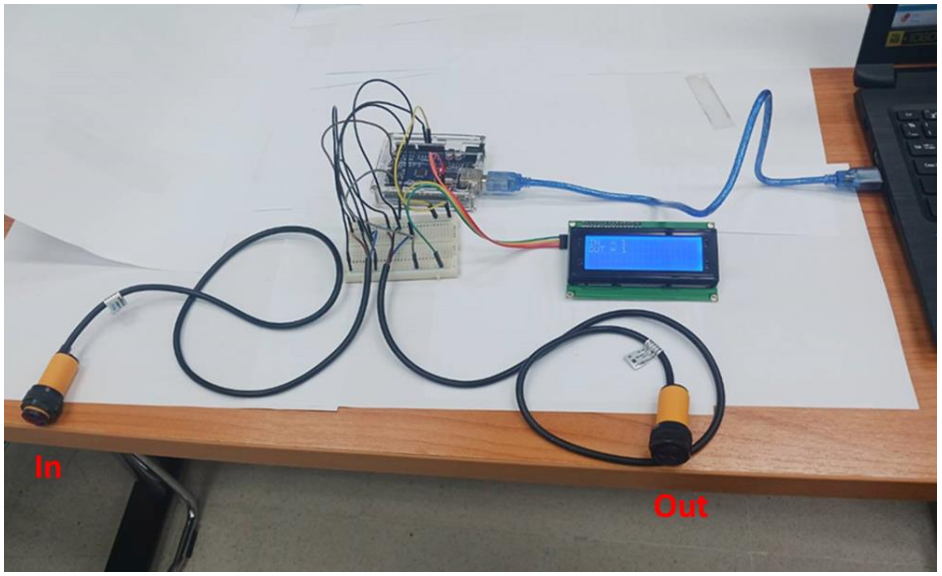
```
  pinMode(IR_OUT, INPUT); // กำหนดขาของเซ็นเซอร์อินฟราเรดเป็นขาอินพุต
}
void loop() {
  checkWalkIn(); // เรียกใช้ฟังก์ชันเพื่อตรวจสอบการเข้า

  checkWalkOut(); // เรียกใช้ฟังก์ชันเพื่อตรวจสอบการออก


  if (isWalkIn && isWalkOut) { // หากมีการเข้าและออกพร้อมกัน
    lcdUpdate(); // เรียกใช้ฟังก์ชันเพื่ออัปเดตข้อมูลบนจอ LCD
    isWalkIn = false; // รีเซ็ตสถานะการเข้า
    isWalkOut = false; // รีเซ็ตสถานะการออก
  }
}


void checkWalkIn() {
  if (last_In != digitalRead(IR_IN)) { // ตรวจสอบการเปลี่ยนแปลงของสถานะของเซ็นเซอร์
    last_In = digitalRead(IR_IN); // บันทึกสถานะปัจจุบันของเซ็นเซอร์
// ตรวจสอบเงื่อนไขการเข้า

    if (digitalRead(IR_IN) == LOW && isWalkIn == false && digitalRead(IR_OUT) == HIGH) {
      walkInTime = millis(); // บันทึกเวลาเมื่อเริ่มต้นการเข้า
      isWalkIn = true; // บันทึกสถานะการเข้า
    }
  }
}
void checkWalkOut() {
  if (last_Out != digitalRead(IR_OUT)) { // ตรวจสอบการเปลี่ยนแปลงของสถานะของเซ็นเซอร์
    last_Out = digitalRead(IR_OUT); // บันทึกสถานะปัจจุบันของเซ็นเซอร์
// ตรวจสอบเงื่อนไขการออก

    if (digitalRead(IR_OUT) == LOW && isWalkOut == false && digitalRead(IR_IN) == HIGH) {
      walkOutTime = millis(); // บันทึกเวลาเมื่อเริ่มต้นการออก
      isWalkOut = true; // บันทึกสถานะการออก
    }
```

```
  }
}
void lcdUpdate() {
  if (walkInTime - walkOutTime < 0) { // ถ้าเวลาการเข้ามากกว่าเวลาการออก
    counter_in++; // เพิ่มจำนวนคนที่เข้า
    lcd.setCursor(6, 0); // ตั้งตำแหน่ง cursor ที่คอลัมน์ 6 และแถว 0
    lcd.print(counter_in); // แสดงจำนวนคนที่เข้าบนจอ LCD
  }
  if (walkOutTime - walkInTime < 0) { // ถ้าเวลาการออกมากกว่าเวลาการเข้า
    counter_out++; // เพิ่มจำนวนคนที่ออก
    lcd.setCursor(6, 1); // ตั้งตำแหน่ง cursor ที่คอลัมน์ 6 และแถว 1

    lcd.print(counter_out); // แสดงจำนวนคนที่ออกบนจอ LCD
  }
}
```

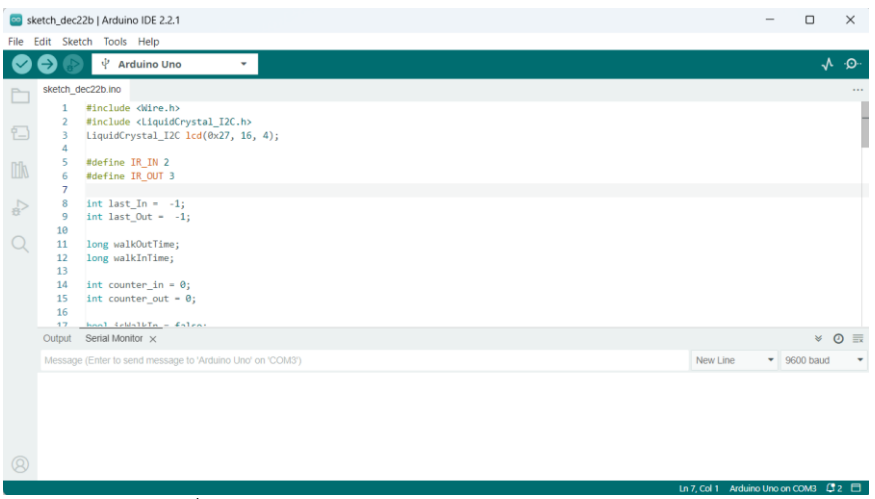
ให้ทำการเขียนโปรแกรมข้างต้น ในโปรแกรม Arduino IDE แล้วทำการอัปโหลดโปรแกรมลงบอร์ด Arduino UNO ก็จะสามารถทำงานได้ตรงตามโปรแกรมที่กำหนดเรียบร้อย ดังรูปที่ 5.33



รูปที่ 5.31 การต่อวงจรเซ็นเซอร์อินฟราเรด และ LCD บนบอร์ด Arduino UNO



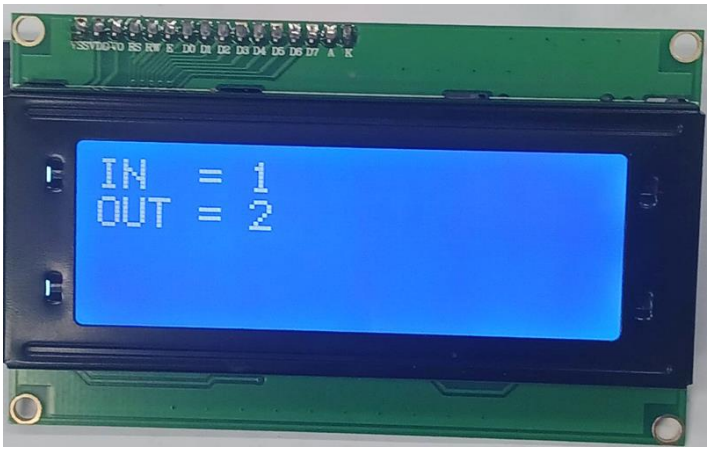
รูปที่ 5.32 การติดตั้งอุปกรณ์ขาเข้า และขาออก



รูปที่ 5.33 การอัปโหลดโปรแกรมลงบอร์ด Arduino

5.5.3 ผลลัพธ์ที่ได้

เมื่อมีการคนเดินทางผ่านเข้า In และผ่านมาที่ Out จะแสดงว่ามีการเดินทางเข้า ทำนองเดียวกันหากมีการเดินทางผ่านจาก Out และผ่านไปที่ In จะแสดงว่ามีจำนวนคนเดินออก ซึ่งจะแสดงยอดจำนวนคนเข้าและออกแสดงดังรูปที่ 5.34



รูปที่ 5.34 ผลลัพธ์จำนวนคนเข้าและออกที่นับได้บนจอ LCD1602

สรุปท้ายบท

สำหรับการใช้งานในโพรเจกต์ในรูปแบบต่างๆ ที่นำเสนอจะขึ้นอยู่กับว่าผู้ออกแบบระบบต้องการใช้อุปกรณ์เซ็นเซอร์แบบใดบ้างในการควบคุมการทำงานผ่านระบบอัตโนมัติ โดยการใช้บอร์ด Arduino ซึ่งจะเห็นว่า มีการใช้สมาร์ตคาร์ด RFID มาประยุกต์ใช้งาน ตรวจสอบ และให้ระบบทำงานตาม RFID หมายเลขที่กำหนดได้ การใช้เซ็นเซอร์ตรวจวัดอุณหภูมิ ความชื้น ตลอดจนการใช้เซ็นเซอร์อินฟราเรดในการตรวจสอบคนเข้า และคนออก โดยทั้งหมดสามารถแสดงผลผ่านหน้าจอ LCD ได้ ซึ่งเป็นการประยุกต์ Arduino สำหรับการใช้งานและโพรเจกต์ที่ต้องการ ในหลากหลายสถานการณ์ที่เกิดขึ้น

คำถามท้ายบท

1. ให้แสดงการใช้งานบอร์ด Arduino เชื่อมต่อกับ RFID พร้อมเขียนโปรแกรม และอธิบายพอสังเขป

2. ให้แสดงการใช้งานบอร์ด Arduino เชื่อมต่อกับ RFID เพื่อใช้ในการเปิด หรือปิดไฟ พร้อมเขียนโปรแกรม และอธิบายพอสังเขป

3. ให้แสดงการใช้งานบอร์ด Arduino เชื่อมต่อกับ DHT22 ในการวัดอุณหภูมิ ความชื้น พร้อมแสดงผลผ่านจอ LCD1602 พร้อมเขียนโปรแกรม และอธิบายพอสังเขป

4. โมดูล Smart Card reader ที่นิยมใช้กับ Arduino มีชื่อว่าอะไร

5. ไลบรารีใดที่จำเป็นต้องติดตั้งเพื่อใช้งานเซ็นเซอร์ DHT กับ Arduino

เอกสารอ้างอิง

1. เดชฤทธิ์ มณีธรรม, *คัมภีร์การใช้งานไมโครคอนโทรลเลอร์ Arduino*, กรุงเทพฯ: ซีเอ็ดยูเคชั่น, 2560.

2. สุวิทย์ เมาะราษี, *การโปรแกรมมิ่งบอร์ด Arduino ด้วย C# .NET และ Sketch*, สำนักพัฒนาสมรรถนะครูและบุคลากรอาชีวศึกษา, 2562.

3. "*Arduino - Home*," Arduino.cc. [Online]. Available: https://www.arduino.cc/. ,2023.

4. S. Fitzgerald and M. Shiloh, *The Arduino Projects Book*, Torino, Italy, 2012.

5. J. Blum, *Exploring Arduino: Tools and Techniques for Engineering Wizardry*, Wiley, 2013.

6. โพรเจกต์โรงประลองต้นแบบทางวิศวกรรม, *เอกสารประกอบคำสอนเรื่อง Computer and Electronics*, มหาวิทยาลัยธรรมศาสตร์, 2562.

7. ทศพล บ้านคลองสี่, *Practical Microcontroller Programming with Arduino*, กรุงเทพฯ: ไอดีซีพรีเมียร์, 2565.

บทที่ 6

กรณีศึกษา โพรเจกต์งานวิจัย I

การเรียนรู้พื้นฐานที่ผ่านมา สามารถที่จะทำให้เกิดความเข้าใจเกี่ยวกับการประยุกต์ Arduino ในการทำงานต่างๆ ที่ใช้เซ็นเซอร์พื้นฐานที่มีลักษณะแตกต่างกัน เพื่อให้เข้าใจการทำงานในลักษณะต่างๆ สำหรับในสองบทสุดท้ายนี้ จะเป็นการประยุกต์ Arduino ที่ใช้สำหรับงานวิจัย ที่มีความลุ่มลึกทางวิชาการขึ้นไป ซึ่งจะทำให้ผู้อ่านเข้าใจการประยุกต์ใช้งานในเชิงการวิจัยได้ดียิ่งๆขึ้นไป

6.1 เครื่องจ่ายเจลแอลกอฮอล์ล้างมืออัตโนมัติด้วยเซ็นเซอร์เหนือเสียงและมอเตอร์เซอร์โว

การแพร่ระบาดของเชื้อไวรัส COVID-19 ได้สร้างความท้าทายอย่างมากให้กับสังคมทั่วโลก ทำให้ต้องมีการนำมาตรการต่างๆ มาใช้เพื่อลดการแพร่กระจายของเชื้อ หนึ่งในมาตรการที่ไม่ใช่การใช้ยา (Non-pharmacological interventions) ที่สำคัญได้แก่ การรักษาสุขอนามัยของมือ การสวมหน้ากากอนามัย การใช้เฟซชิลด์ และการเว้นระยะห่างทางสังคม จากองค์การอนามัยโลกพบว่าการรักษาสุขอนามัยของมือเป็นวิธีที่สำคัญในการป้องกันการแพร่กระจายของเชื้อ COVID-19 และการล้างมืออย่างสม่ำเสมอสามารถลดความเสี่ยงในการแพร่เชื้อไวรัสได้ถึง 55% [1-4] การปฏิบัติตามการรักษาสุขอนามัยของมือถือเป็นการกระทำที่สำคัญในการป้องกันการแพร่กระจายของเชื้อโรคอย่างแพร่หลาย และมีงานวิจัยหลายฉบับที่ได้กล่าวถึงการพัฒนาเครื่องจ่ายน้ำยาฆ่าเชื้ออัตโนมัติ ซึ่งสามารถทำให้เป็นอัตโนมัติแบบไม่ต้องสัมผัสได้โดยใช้เซ็นเซอร์หลายประเภทเพื่อตรวจจับความใกล้เคียง [5-9] โดยปกติแล้วเครื่องจ่ายน้ำยาฆ่าเชื้อราคาประหยัดมักใช้เซ็นเซอร์อินฟราเรด (IR) [10-12] อย่างไรก็ตาม เซ็นเซอร์อินฟราเรดมักทำงานไม่ดีในที่ที่มีเสียงรบกวนมากหรือในสถานที่ที่มีการเปลี่ยนแปลงของแสงแดดเช่นในวันที่มีเมฆคลุมหรือเมื่อมีการสะท้อนแสงจากพื้นดิน ในขณะที่เซ็นเซอร์เหนือเสียงสามารถทำงานได้ดีกว่าในสภาพแวดล้อมภายนอก[13-14]

ในงานวิจัยนี้ [15] ผู้วิจัยได้ทำการศึกษาและพัฒนาเครื่องจ่ายแอลกอฮอล์ล้างมืออัตโนมัติโดยใช้เซ็นเซอร์เหนือเสียงและมอเตอร์เซอร์โวที่มหาวิทยาลัย ซึ่งเครื่องจ่ายนี้มีประโยชน์ดังนี้

1. สะดวกในการใช้งาน

2. อุปกรณ์ไม่ต้องสัมผัสโดยทำงานอัตโนมัติ

3. หาอุปกรณ์ได้ง่าย

งานวิจัยนี้มีเป้าหมายเพื่อพัฒนาและปรับปรุงเครื่องจ่ายแอลกอฮอล์ล้างมือที่มีประสิทธิภาพและเหมาะสมกับการใช้งานในสภาพแวดล้อมต่างๆ โดยเฉพาะในที่สาธารณะ

6.2 วิธีการวิจัย

การศึกษานี้ใช้ระเบียบวิธีวิจัยที่แบ่งออกเป็นสี่ขั้นตอนหลัก ดังแสดงในรูปที่ 6.1 ประกอบด้วยดังนี้

1. การออกแบบ (Design)

- ขั้นตอนแรกคือการออกแบบระบบเครื่องจ่ายแอลกอฮอล์ล้างมืออัตโนมัติที่ใช้เซ็นเซอร์เหนือเสียงและมอเตอร์เซอร์โว โดยการเลือกอุปกรณ์และองค์ประกอบต่าง ๆ ที่เหมาะสม รวมถึงการกำหนดรูปแบบและขนาดของอุปกรณ์เพื่อให้สอดคล้องกับความต้องการใช้งานในสภาพแวดล้อมที่แตกต่างกัน

2. การพัฒนาและติดตั้ง (Implement)

- ขั้นตอนนี้คือการนำแบบที่ออกแบบมาไปพัฒนาและติดตั้งจริง โดยการประกอบและเชื่อมต่ออุปกรณ์ต่าง ๆ เข้าด้วยกัน รวมถึงการเขียนโปรแกรมควบคุมมอเตอร์เซอร์โวและการตั้งค่าการตรวจจับของเซ็นเซอร์เหนือเสียง เพื่อให้ระบบสามารถทำงานได้อย่างมีประสิทธิภาพและตอบสนองต่อการใช้งานจริง

3. การทดสอบและผลลัพธ์ (Result)

- เมื่อพัฒนาระบบเสร็จสิ้น ระบบจะถูกทดสอบในสภาพแวดล้อมที่หลากหลายเพื่อประเมินประสิทธิภาพและความสามารถในการทำงาน ซึ่งจะรวมถึงการวัดความแม่นยำของการตรวจจับโดยเซ็นเซอร์เหนือเสียง การตอบสนองของมอเตอร์เซอร์โว และการประเมินความสะดวกในการใช้งานของผู้ใช้

4. การสำรวจความพึงพอใจ (Survey Satisfaction)

- หลังจากการทดสอบเบื้องต้น จะมีการสำรวจความพึงพอใจของผู้ใช้ที่ทดลองใช้ระบบ โดยการเก็บข้อมูลจากการสอบถามความคิดเห็นและข้อเสนอแนะของผู้ใช้จริง ซึ่งข้อมูลเหล่านี้จะถูกนำมาวิเคราะห์เพื่อปรับปรุงและพัฒนาระบบให้ดียิ่งขึ้นในอนาคต

รูปที่ 6.1 วิธีการวิจัย

6.2.1 การออกแบบ

สามารถแบ่งออกเป็นสี่ขั้นตอน แสดงดังรูปที่ 6.2 ดังนี้

6.2.1.1 การออกแบบ (Design)

สถาปัตยกรรมของระบบเครื่องจ่ายแอลกอฮอล์ล้างมืออัตโนมัติประกอบด้วยสามส่วนหลัก ดังแสดงในรูปที่ 6.2

1. เซ็นเซอร์เหนือเสียง(Ultrasonic Sensor)

- เมื่อผู้ใช้วางมือใต้เซ็นเซอร์ เซ็นเซอร์เหนือเสียงจะตรวจจับตำแหน่งของมือและส่งข้อมูลไปยังบอร์ด Arduino

- เซ็นเซอร์เหนือเสียงทำงานโดยการส่งคลื่นเสียงและวัดระยะเวลาที่คลื่นเสียงสะท้อนกลับมา ทำให้สามารถตรวจจับระยะห่างระหว่างเซ็นเซอร์กับมือของผู้ใช้ได้อย่างแม่นยำ

2. บอร์ด Arduino

- บอร์ด Arduino จะรับสัญญาณจากเซ็นเซอร์เหนือเสียง เมื่อพบการตรวจจับมือของผู้ใช้ บอร์ด Arduino จะสั่งให้มอเตอร์เซอร์โวทำงาน

- บอร์ด Arduino จะถูกตั้งโปรแกรมให้รับข้อมูลจากเซ็นเซอร์และประมวลผลเพื่อควบคุมการทำงานของมอเตอร์เซอร์โวอย่างเหมาะสม

3. มอเตอร์เซอร์โว (Servo Motor)

- บอร์ด Arduino จะส่งสัญญาณไปยังมอเตอร์เซอร์โวเพื่อให้หมุนแกนที่ 40 องศา ทำให้มีการปล่อยเจลแอลกอฮอล์ในปริมาณที่กำหนดออกมาจากหัวฉีด

ขั้นตอนการทำงานของระบบ

1. การตรวจจับ (Detection)

- ผู้ใช้วางมือใต้หัวฉีดและเซ็นเซอร์เหนือเสียงจะตรวจจับระยะห่าง

- เซ็นเซอร์จะส่งสัญญาณไปยังบอร์ด Arduino

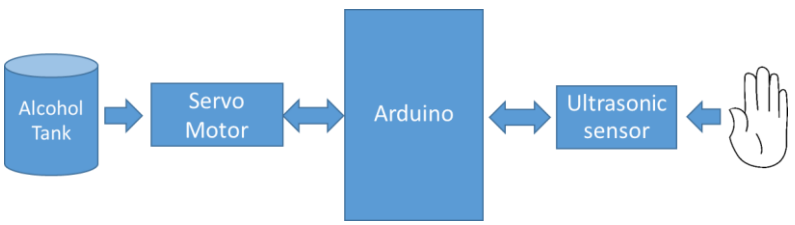
2. การประมวลผล (Processing)

- บอร์ด Arduino รับสัญญาณจากเซ็นเซอร์และประมวลผลเพื่อตรวจสอบว่ามือของผู้ใช้อยู่ในระยะที่กำหนด

3. การจ่ายแอลกอฮอล์ (Dispensing)

- เมื่อบอร์ด Arduino ยืนยันว่ามีการตรวจจับมือของผู้ใช้ จะส่งสัญญาณไปยังมอเตอร์เซอร์โว
- มอเตอร์เซอร์โวจะหมุนที่ 40 องศาเพื่อจ่ายแอลกอฮอล์ในปริมาณที่กำหนดออกมาจากหัวฉีด

ระบบนี้ถูกออกแบบให้มีประสิทธิภาพในการใช้งานและสามารถใช้ในสภาพแวดล้อมต่างๆ ได้อย่างดี เยี่ยมโดยเฉพาะในสถานที่ที่มีคนใช้มากและต้องการมาตรการสุขอนามัยที่เข้มงวด



รูปที่ 6.2 โครงสร้างการออกแบบระบบ

6.2.2 อุปกรณ์ที่ใช้

1.เซ็นเซอร์เหนือเสียง

เซ็นเซอร์เหนือเสียง เป็นอุปกรณ์ที่ใช้คลื่นเสียงความถี่สูง (โดยปกติจะสูงกว่า 20 kHz ซึ่งสูงเกินกว่าที่มนุษย์สามารถได้ยิน) เพื่อวัดระยะทางหรือการตรวจจับวัตถุต่างๆ เซ็นเซอร์ประเภทนี้ทำงานโดยส่งคลื่นเสียงไปยังเป้าหมายและรับสัญญาณสะท้อนกลับจากเป้าหมาย ซึ่งจากนั้นจะใช้เวลาที่คลื่นเสียงเดินทางไปกลับระหว่างเซ็นเซอร์และวัตถุในการคำนวณระยะทาง

หลักการทำงาน

1. การส่งสัญญาณ (Transmission) เซ็นเซอร์จะปล่อยคลื่นเสียงเหนือเสียงออกไปในทิศทางที่กำหนด
2. การสะท้อนกลับ (Echo) เมื่อคลื่นเสียงชนกับวัตถุ มันจะสะท้อนกลับไปยังเซ็นเซอร์
3. การรับสัญญาณ (Reception) เซ็นเซอร์จะจับสัญญาณสะท้อนกลับที่เข้ามา
4. การคำนวณระยะทาง (Distance Calculation) โดยใช้เวลาที่คลื่นเสียงเดินทางไปกลับ (ไปและกลับจากวัตถุ) และความเร็วของเสียงในอากาศ (ประมาณ 343 เมตร/วินาทีในอุณหภูมิห้อง) เซ็นเซอร์จะคำนวณระยะทางถึงวัตถุโดยใช้สูตร

$$\text{ระยะทาง} = \frac{\text{เวลาเดินทาง}}{2} \times \text{ความเร็วเสียง}$$

เนื่องจากคลื่นเสียงเดินทางไปและกลับ จึงต้องหารเวลาที่ได้มาด้วย 2

การใช้งานของเซ็นเซอร์เหนือเสียง

- ระบบอัตโนมัติทางอุตสาหกรรม ใช้ในการวัดระดับของเหลวหรือวัตถุในถัง

- ระบบรักษาความปลอดภัย ตรวจจับการเคลื่อนไหวหรือผู้บุกรุก

- ยานพาหนะ ระบบช่วยจอดรถที่ใช้เพื่อเตือนผู้ขับขี่เมื่อมีวัตถุใกล้เข้ามา

- หุ่นยนต์ ใช้เพื่อวัดระยะทางถึงวัตถุหรือหลีกเลี่ยงการชน

ข้อดีและข้อเสียของเซ็นเซอร์เหนือเสียง

ข้อดี

- ไม่ขึ้นกับแสงสว่าง สามารถทำงานได้ในสภาพแสงที่หลากหลาย รวมถึงในที่มืด

- ความแม่นยำ มีความแม่นยำสูงในการวัดระยะทางถึงวัตถุ

- ไม่ต้องการการสัมผัส ทำงานได้โดยไม่ต้องมีการสัมผัสกับวัตถุ

ข้อเสีย

- ผลกระทบจากอุณหภูมิและความชื้น ความเร็วของเสียงอาจเปลี่ยนแปลงตามอุณหภูมิและความชื้นของอากาศ ซึ่งอาจทำให้การวัดไม่แม่นยำ

- การสะท้อนจากพื้นผิวที่ซับซ้อน วัตถุที่มีพื้นผิวโค้งหรือดูดซับเสียงอาจทำให้การสะท้อนกลับไม่ดี ส่งผลต่อความแม่นยำในการวัด

- ช่วงการวัดจำกัด ระยะการตรวจจับที่ใช้งานได้จริงจำกัดโดยขึ้นอยู่กับความถี่และกำลังของเซ็นเซอร์

เซ็นเซอร์เหนือเสียงเป็นเครื่องมือที่มีความหลากหลายในการใช้งานและมีความแม่นยำสูงในงานที่ต้องการการวัดระยะทางหรือการตรวจจับวัตถุในระยะใกล้ถึงกลาง เซ็นเซอร์นี้ถูกนำไปใช้ในหลายอุตสาหกรรม ตั้งแต่การผลิตจนถึงการใช้งานในชีวิตประจำวัน เช่น รถยนต์และหุ่นยนต์ เป็นต้น เซ็นเซอร์เหนือเสียง HC-SR04 แสดงได้ดังรูปที่ 6.3

รูปที่ 6.3 เซ็นเซอร์เหนือเสียง

2.บอร์ด Arduino UNO

ในการวิจัยครั้งนี้ ได้ทำการเลือกใช้บอร์ด Arduino UNO ซึ่งแสดงดังรูปที่ 6.4

รูปที่ 6.4 บอร์ด Arduino UNO

3. เซอร์โวมอเตอร์ (Servo Motor)

เซอร์โวมอเตอร์ เป็นมอเตอร์ชนิดหนึ่งที่มีการควบคุมตำแหน่ง ความเร็ว หรือแรงบิดอย่างแม่นยำและรวดเร็ว มอเตอร์นี้มักประกอบด้วยส่วนหลัก 3 ส่วน ได้แก่

1. มอเตอร์ไฟฟ้า (Electric Motor) ที่ทำหน้าที่สร้างการหมุนหรือการเคลื่อนไหว

2. ตัวควบคุม (Controller) ที่ทำหน้าที่ควบคุมการทำงานของมอเตอร์ โดยใช้สัญญาณคำสั่งที่ได้รับจากระบบควบคุม

3. ตัวเข้ารหัส (Encoder) หรือ เครื่องวัดตำแหน่ง (Potentiometer) ที่ให้ข้อมูลกลับไปยังตัวควบคุมเกี่ยวกับตำแหน่งปัจจุบันของมอเตอร์ ทำให้สามารถปรับการเคลื่อนไหวได้อย่างแม่นยำ

หลักการทำงานของเซอร์โวมอเตอร์

1. การรับคำสั่ง (Receiving Command) เซอร์โวมอเตอร์จะได้รับสัญญาณคำสั่งที่บอกตำแหน่งหรือความเร็วที่ต้องการ

2. การปรับความเคลื่อนไหว (Adjusting Movement) ตัวควบคุมจะปรับความเคลื่อนไหวของมอเตอร์เพื่อให้ได้ผลลัพธ์ตามคำสั่ง โดยการหมุนมอเตอร์จนกว่าตำแหน่งหรือความเร็วจะตรงกับสัญญาณคำสั่ง

3. การตรวจสอบตำแหน่ง (Position Feedback) เครื่องวัดตำแหน่งจะส่งข้อมูลตำแหน่งปัจจุบันกลับไปยังตัวควบคุม เพื่อให้มั่นใจว่ามอเตอร์อยู่ในตำแหน่งที่ถูกต้อง

การใช้งาน

เซอร์โวมอเตอร์ถูกใช้งานในหลายด้าน เช่น

- หุ่นยนต์ สำหรับควบคุมการเคลื่อนไหวที่แม่นยำของแขนหรือข้อต่อ

- อุปกรณ์อัตโนมัติในอุตสาหกรรม ใช้ในการควบคุมเครื่องจักรที่ต้องการความแม่นยำสูง

- ระบบควบคุมการบิน ใช้ในเครื่องบินและโดรนสำหรับการควบคุมทิศทางและการเคลื่อนที่

ข้อดี

- ความแม่นยำสูง สามารถควบคุมตำแหน่งและความเร็วได้อย่างแม่นยำ

- ตอบสนองรวดเร็ว ปรับการเคลื่อนไหวได้รวดเร็วตามคำสั่ง

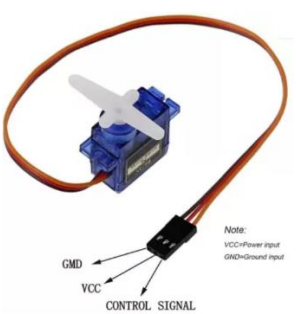
- ความน่าเชื่อถือสูง ทนทานและมีความเสถียร

ข้อเสีย

- ราคาสูง มักมีราคาสูงกว่าเมื่อเทียบกับมอเตอร์ทั่วไป

- ต้องการการตั้งค่า การติดตั้งและปรับจูนต้องการความรู้และความชำนาญ

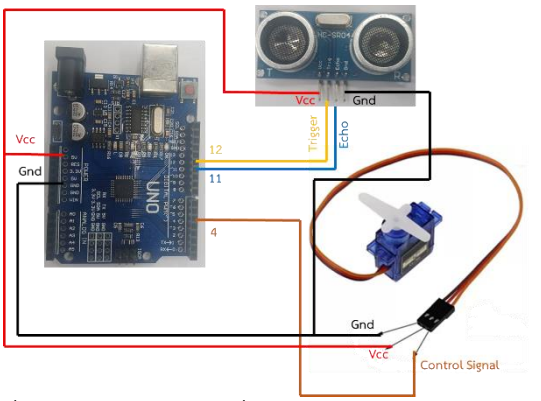
เซอร์โวมอเตอร์เป็นเครื่องมือที่สำคัญและขาดไม่ได้ในการควบคุมการเคลื่อนไหวในหลายระบบที่ต้องการความแม่นยำสูงและการตอบสนองที่รวดเร็ว แสดงได้ดังรูปที่ 6.5



https://www.flipkart.com/super-debug-sg-90-tower-pro-micro-servo-motor-electronic-components-hobby-kit/p/itmfcd25893fba20: Accessed: Jul.2, 2024.

รูปที่ 6.5 Servo Motor

ให้ทำการต่อวงจรดังรูปที่ 6.6



รูปที่ 6.6 การต่อวงจรเครื่องจ่ายแอลกอฮอล์อัตโนมัติ

6.2.3 การเขียนโปรแกรมสำหรับเครื่องจ่ายแอลกอฮอล์

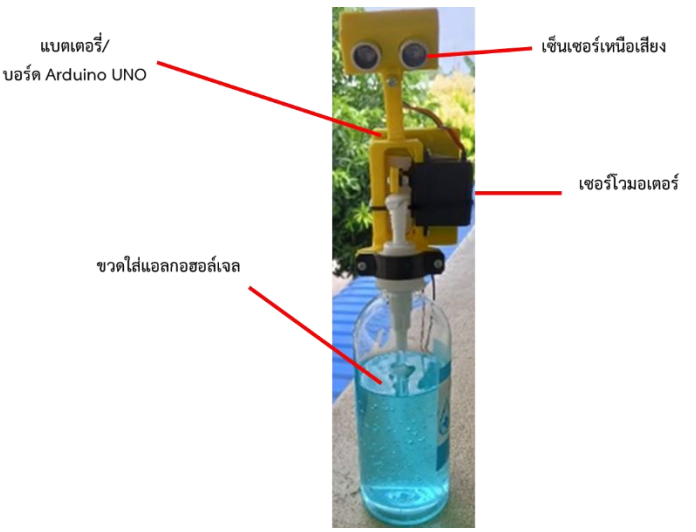
ให้เขียนโปรแกรมตามคำสั่งด้านล่างนี้

```
#include<NewPing.h> // ไลบรารีสำหรับการใช้งานเซ็นเซอร์เหนือเสียง

#include <Servo.h>  // ไลบรารีสำหรับการควบคุมเซอร์โวมอเตอร์

// กำหนดขาสำหรับเซ็นเซอร์เหนือเสียง

#define TRIGGER_PIN  12 // ขาที่ 12 สำหรับส่งสัญญาณ Trigger

#define ECHO_PIN     11 // ขาที่ 11 สำหรับรับสัญญาณ Echo

#define MAX_DISTANCE 20 // ระยะทางสูงสุดที่เซ็นเซอร์จะวัดได้ 20 เซนติเมตร

// สร้างอ็อบเจ็กต์สำหรับเซ็นเซอร์เหนือเสียง

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);

// สร้างอ็อบเจ็กต์สำหรับเซอร์โวมอเตอร์

Servo myservo;

int state = 0; // ตัวแปรสำหรับเก็บสถานะการทำงาน

void setup() {

  Serial.begin(115200); // เริ่มการสื่อสารผ่าน Serial เพื่อการแสดงผล
```

```
  myservo.attach(4);    // เชื่อมต่อเซอร์โวมอเตอร์เข้ากับขาดิจิตอลหมายเลข 4
  myservo.write(40);    // ตั้งค่ามุมเริ่มต้นของเซอร์โวมอเตอร์ที่ 40 องศา
}
void loop() {
  delay(500); // หน่วงเวลาการทำงาน 500 มิลลิวินาที เพื่อให้การทำงานเป็นไปอย่างเสถียร
  int SR = sonar.ping_cm(); // วัดระยะทางจากเซ็นเซอร์เหนือเสียงในหน่วยเซนติเมตร
  Serial.println(SR); // แสดงผลระยะทางที่วัดได้ผ่าน Serial Monitor

  // ตรวจสอบสถานะการทำงาน
  if (state == 0) {
    if (SR <= 1) { // ถ้าระยะทางที่วัดได้เท่ากับหรือน้อยกว่า 1 ซม.
      myservo.write(40); // ตั้งค่ามุมเซอร์โวที่ 40 องศา
    }
    else {
      delay(700); // หน่วงเวลา 700 มิลลิวินาที
      myservo.write(0); // หมุนเซอร์โวไปที่ 0 องศา
      delay(700); // หน่วงเวลาอีก 700 มิลลิวินาที
      state = 1; // เปลี่ยนสถานะไปที่ 1
    }
  }
  else if (state == 1) {
    myservo.write(40); // ตั้งค่ามุมเซอร์โวที่ 40 องศา
    delay(5000); // หน่วงเวลาการทำงาน 5 วินาที
```

```
    state = 0; // เปลี่ยนสถานะกลับไปที่ 0
  }
}
```

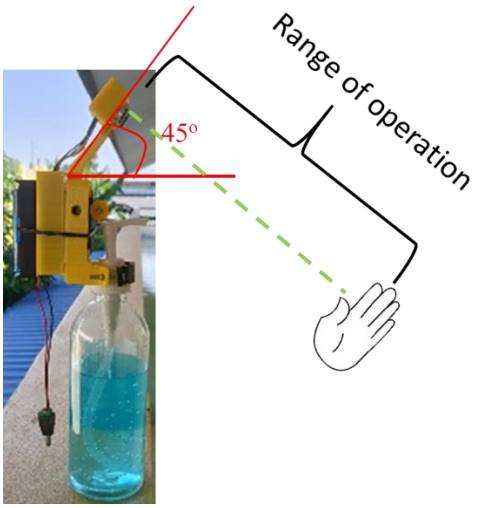
เมื่อเขียนคำสั่งข้างต้นเสร็จสิ้น ให้ทำการอัปโหลดโปรแกรมให้ Arduino ทำงานตามที่กำหนด ส่วนประกอบของเครื่องมือจ่ายแอลกอฮอล์อัตโนมัติแสดงได้ดังรูปที่ 6.7



รูปที่ 6.7 ส่วนประกอบต่างๆ ในเครื่องจ่ายแอลกอฮอล์อัตโนมัติ

6.3 ผลการทดลอง

การทดสอบระยะการทำงานของเซ็นเซอร์ จะวัดโดยแสดงได้ดังรูปที่ 6.8 แล้วทำการทดสอบโดยผลที่ได้จะแสดงได้ดังตารางที่ 6.1



รูปที่ 6.8 ช่วงทดสอบระยะการทำงานของเครื่องจ่ายแอลกอฮอล์อัตโนมัติ

ตารางที่ 6.1 ผลการวัดระยะการทำงานของเครื่องจ่ายแอลกอฮอล์อัตโนมัติ

	ระยะการทำงาน						
ระยะห่างระหว่างฝ่ามือกับเซ็นเซอร์	1 ซ.ม.	2 ซ.ม.	5 ซ.ม.	10 ซ.ม.	15 ซ.ม.	20 ซ.ม.	25 ซ.ม.
ครั้งที่							
1	ไม่ทำงาน	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ไม่ทำงาน	ไม่ทำงาน
2	ไม่ทำงาน	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ไม่ทำงาน
3	ไม่ทำงาน	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ไม่ทำงาน	ไม่ทำงาน
4	ไม่ทำงาน	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ไม่ทำงาน	ไม่ทำงาน
5	ไม่ทำงาน	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ไม่ทำงาน
6	ไม่ทำงาน	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ไม่ทำงาน
7	ไม่ทำงาน	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ไม่ทำงาน
8	ไม่ทำงาน	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ไม่ทำงาน	ไม่ทำงาน
9	ไม่ทำงาน	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ไม่ทำงาน
10	ไม่ทำงาน	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ไม่ทำงาน
ค่าเฉลี่ยเปอร์เซ็นต์การทำงาน	0%	100%	100%	100%	100%	60%	40%

จากตารางข้างต้น จะพบว่าระยะการทำงานที่เหมาะสมระหว่างมือของผู้ใช้และเซ็นเซอร์เหนือเสียงอยู่ที่ 2 ซม., 5 ซม., 10 ซม., และ 15 ซม. โดยผลการทำงานที่ได้แสดงประสิทธิภาพในการตรวจจับที่ 100% ในระยะเหล่านี้

สรุปท้ายบท

1. ระยะการตรวจจับที่เหมาะสม

- ระยะที่เครื่องสามารถตรวจจับและทำงานได้อย่างแม่นยำคือระหว่าง 2 ซม. ถึง 15 ซม. ที่ระดับนี้ เครื่องจ่ายแอลกอฮอล์สามารถทำงานได้เต็มประสิทธิภาพ โดยจ่ายแอลกอฮอล์ออกมาในปริมาณที่กำหนดทันทีที่ตรวจจับได้ว่ามือของผู้ใช้อยู่ในระยะที่กำหนด

2. ระยะการทำงานที่ลดลง

- เมื่อระยะระหว่างมือและเซ็นเซอร์เพิ่มขึ้นเป็น 20 ซม. ประสิทธิภาพในการทำงานลดลงเหลือเพียง 60% และที่ระยะ 25 ซม. ประสิทธิภาพลดลงเหลือเพียง 40% ซึ่งบ่งชี้ว่าเซ็นเซอร์เหนือเสียงมีขีดจำกัดในการตรวจจับที่มีความแม่นยำน้อยลงเมื่อระยะทางระหว่างมือและเซ็นเซอร์เพิ่มขึ้น

3. การไม่ทำงานที่ระยะใกล้มาก

- ที่ระยะ 1 ซม. เครื่องจ่ายแอลกอฮอล์ไม่สามารถทำงานได้เลย ซึ่งอาจเกิดจากข้อจำกัดทางกายภาพของเซ็นเซอร์เหนือเสียงในการตรวจจับวัตถุที่อยู่ใกล้มากเกินไป

คำถามท้ายบท

1.ให้อธิบายวิธีการวิจัยที่ใช้ในงานวิจัยนี้ พอสังเขป

2. จงอธิบายการทำงานของเซ็นเซอร์เหนือเสียง พอสังเขป

3. จงอธิบายการทำงานของเซอร์โวมอเตอร์ พอสังเขป

4. จงอธิบายการออกแบบเครื่องจ่ายเจลแอลกอฮอล์ล้างมืออัตโนมัติด้วยเซ็นเซอร์เหนือเสียงและมอเตอร์เซอร์โว พอสังเขป

5. ท่านคิดว่าจะนำการอุปกรณ์ที่ใช้ในงานวิจัยนี้ ไปพัฒนาเป็นงานวิจัยเรื่องใด โปรดระบุการนำไปใช้งานที่ท่านต้องการ พอสังเขป

เอกสารอ้างอิง

1. N. Ferguson, D. Laydon, G. Nedjati Gilani, N. Imai, K. Ainslie, M. Baguelin, S. Bhatia, A. Boonyasiri, Z. Cucunuba Perez, and G. Cuomo-Dannenburg, “Report 9: Impact of non-pharmaceutical interventions (NPIs) to reduce COVID19 mortality and healthcare demand,” 2020.

2. R. Guner, I. Hasanoglu, and F. Aktas, “COVID-19: Prevention and control measures in community,” Turk J Med Sci, vol. 50, SI-1, pp. 571-577, 2020.

3. T. Jefferson, R. Foxlee, C. Del Mar, L. Dooley, E. Ferroni, B. Hewak, A. Prabhala, S. Nair, and A. Rivetti, “Physical interventions to interrupt or reduce the spread of respiratory viruses: systematic review,” BMJ, vol. 336, no. 7635, pp. 77-80, 2008.

4. S. P. Luby, M. Agboatwalla, D. R. Feikin, J. Painter, W. Billhimer, A. Altaf, and R. M. Hoekstra, “Effect of handwashing on child health: a randomised controlled trial,” The Lancet, vol. 366, no. 9481, pp. 225-233, 2005.

5. X. Chen, L. Ran, Q. Liu, Q. Hu, X. Du, and X. Tan, “Hand Hygiene, Mask-Wearing Behaviors and Its Associated Factors during the COVID-19 Epidemic: A Cross-Sectional Study among Primary School Students in Wuhan, China,” Int J Environ Res Public Health, vol. 17, no. 8, 2020.

6. J. M. Boyce, “Measuring healthcare worker hand hygiene activity: current practices and emerging technologies,” Infect Control Hosp Epidemiol, vol. 32, no. 10, pp. 1016-1028, 2011.

7. A. Karn, R. Kanchi, and S. S. Deo, “Design and Development of an Automated Monitored Hand Hygiene System to Curb Infection Spread in Institutional Settings during COVID-19 Pandemic,” 2020.

8. A. Edgar-Tanzil, “Designing for a Pandemic: A Hands-Free Soap Dispeser,” 2020.

9. A. Das, A. Barua, M. A. Mohimin, J. Abedin, M. U. Khandaker, and K. S. Al-mugren, “Development of a Novel Design and Subsequent Fabrication of an Automated Touchless Hand Sanitizer Dispenser to Reduce the Spread of Contagious Diseases,” Healthcare, vol. 9, pp. 1-17, 2021.

10. M. G. Rojo, J. B. Sy, E. R. Calibara, A. V. Comendador, W. Degife, and A. Sisay, “Non-contact temperature reader with sanitizer dispenser (NCTRSD),” Int. J. Sci. Res. Publ., vol. 10, pp. 583-593, 2020.

11. M. M. Srihari, “Self-activating sanitizer with battery imposed system for cleansing hands,” in Proc. 2nd Int. Conf. Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 15–17 July 2020, pp. 1102-1105.

12. Y. Eddy, M. N. Mohammed, I. I. Daoodd, S. H. Bahrain, O. I. Al-Sanjary, and A. K. Sairah, “2019 Novel Coronavirus Disease (COVID-19): Smart contactless hand sanitizer-dispensing system using IoT based robotics technology,” Rev. Argent. Clin. Psicol., vol. 29, pp. 215-220, 2020.

13. S. M. Rabeek, A. Norman, M. Je, M. K. Raja, R. F. Peh, and M. K. Dempsey, “A reliable handwash detector for automated hand hygiene documentation and reminder system in hospitals,” in Proc. 15th Int. Conf. Biomedical Engineering, Springer, Cham, Switzerland, 2014, pp. 892-895.

14. R. Patra, M. Bhattacharya, and S. Mukherjee, “IoT-based computational frameworks in disease prediction and healthcare management: Strategies, challenges, and potential,” in IoT in Healthcare and Ambient Assisted Living, G. Marques, A. K. Bhoi, V. H. C. de Albuquerque, and K. S. Hareesha, Eds., Springer, Singapore, 2021, pp. 17-41.

15. W. Jongchanachavawat, I. Roopkom, N. Mingmuang, C. Wongwian, N. Thanaittipat, V. Chinowan, C. Khiaokhli, and C. Piatuapoo, "*Automatic Alcohol Hand Sanitizer by Using Ultrasonic Sensor and Servo Motor,*" in ECTI-Card 2022.

16. https://www.flipkart.com/super-debug-sg-90-tower-pro-micro-servo-motor-electronic-components-hobby-kit/p/itmfcd25893fba20: Accessed: Jul.2, 2024.

บทที่ 7

กรณีศึกษา โพรเจกต์งานวิจัย II

บทนี้ จะเป็นการประยุกต์ Arduino สำหรับงานวิจัยแบบที่ 2 เป็นเครื่องจ่ายแอลกอฮอล์อัตโนมัติไร้สัมผัสเช่นเดียวกันกับ บทที่ผ่านมา แต่จะมีวิธีการทำงานที่แตกต่างกันออกไป โดยที่จะใช้อุปกรณ์ที่แตกต่างกันกัน ตั้งแต่บอร์ด Arduino ปั๊มมอเตอร์ รีเลย์ เซ็นเซอร์อินฟราเรด เป็นต้น ทำให้ผู้อ่านได้ศึกษาการทำงานของแต่ละอุปกรณ์ ที่จะใช้สำหรับงานวิจัยนี้เพิ่มขึ้น ซึ่งเสมือนเป็นวิธีการทำงานอุปกรณ์ที่แตกต่างกัน ทำให้เกิดความรู้ความเข้าใจในอุปกรณ์ที่เพิ่มขึ้น และทำให้เกิดเข้าใจในการประยุกต์ Arduino สำหรับงานวิชาการที่ลุ่มลึกยิ่งๆ ขึ้นไป

7.1 การประยุกต์ใช้เซ็นเซอร์อินฟราเรดและปั๊มมอเตอร์ในเครื่องจ่ายแอลกอฮอล์อัตโนมัติ

การระบาดของโรค COVID-19 ซึ่งเริ่มขึ้นในเดือนธันวาคม 2019 ที่เมืองอู่ฮั่น ประเทศจีน ได้สร้างความท้าทายครั้งใหญ่ต่อทั่วโลก ไวรัสนี้มีความสามารถในการแพร่กระจายสูง ทำให้องค์การอนามัยโลก (WHO) ต้องออกแนวทางต่างๆ เพื่อช่วยลดการแพร่กระจายในชุมชน หนึ่งในมาตรการที่แนะนำคือการรักษาความสะอาดของมือด้วยการล้างมือบ่อยๆ ด้วยสบู่หรือการใช้เจลแอลกอฮอล์ [1] มีการศึกษาแสดงให้เห็นว่าการล้างมืออย่างสม่ำเสมอสามารถลดความเสี่ยงในการแพร่เชื้อไวรัสได้ประมาณ 55% [3] การรักษาความสะอาดของมือเป็นหนึ่งในมาตรการที่สำคัญที่สุดในการป้องกันการแพร่กระจายของจุลินทรีย์ก่อโรค และสามารถลดการแพร่กระจายของการติดเชื้อ โดยเฉพาะในกลุ่มประชากรที่มีความเสี่ยงสูง [2] เครื่องจ่ายเจลแอลกอฮอล์ล้างมืออัตโนมัติ มีบทบาทสำคัญในการส่งเสริมการรักษาความสะอาดของมือ โดยการจ่ายแอลกอฮอล์ในปริมาณที่เหมาะสมต่อมือ เครื่องจ่ายเจลเหล่านี้ถูกวางไว้ในที่ที่มีการสัญจรหนาแน่น เช่น ห้องน้ำ ทางเดิน ทางเข้าออก และส่วนรับรอง เพื่อให้ผู้ใช้สามารถเข้าถึงได้ง่าย เครื่องจ่ายเจลแอลกอฮอล์มีหลายรูปแบบ รวมถึงแบบที่ทำงานอัตโนมัติเมื่อมีการวางมือใต้เซ็นเซอร์ การทำงานแบบไม่สัมผัสนี้มีประสิทธิภาพสูงในการลดการแพร่กระจายของไวรัสในพื้นที่ที่มีคนหนาแน่น เช่น โรงเรียน มหาวิทยาลัย และห้างสรรพสินค้า [4-5] เครื่องจ่ายเจลแอลกอฮอล์สามารถทำงานได้โดยไม่ต้องสัมผัสและอัตโนมัติผ่านการใช้เซ็นเซอร์ต่างๆ เช่น เซ็นเซอร์อินฟราเรด และเซ็นเซอร์เหนือเสียง [6-7] แต่ละประเภทของเซ็นเซอร์มีข้อดีและข้อจำกัด เซ็นเซอร์อินฟราเรด มักถูกใช้ในเครื่องจ่ายที่มีต้นทุนต่ำ แต่บางครั้งอาจมีปัญหาในสภาพแวดล้อมที่มีแสงสว่างหรือเสียงรบกวนสูง [10-11] งานวิจัยนี้ [15] มุ่งเน้นที่การประยุกต์ใช้เซ็นเซอร์อินฟราเรด ในเครื่องจ่ายแอลกอฮอล์ล้างมืออัตโนมัติ การประเมินเทคโนโลยีเซ็นเซอร์เหล่านี้จะช่วยในการพัฒนาเครื่องจ่ายแอลกอฮอล์ล้างมือที่มีประสิทธิภาพสูงและต้นทุนต่ำ เหมาะสมกับการใช้งานภายในอาคาร

7.2 วิธีการวิจัย

การศึกษานี้ใช้ระเบียบวิธีวิจัยที่แบ่งออกเป็นสี่ขั้นตอนหลัก ดังแสดงในรูปที่ 7.1 ประกอบด้วยดังนี้

1. การออกแบบ (Design)

- ขั้นตอนแรกคือการออกแบบระบบเครื่องจ่ายแอลกอฮอล์ล้างมืออัตโนมัติที่ใช้เซ็นเซอร์อินฟราเรดและปั๊มมอเตอร์ โดยการเลือกอุปกรณ์และองค์ประกอบต่าง ๆ ที่เหมาะสม รวมถึงการกำหนดรูปแบบและขนาดของอุปกรณ์เพื่อให้สอดคล้องกับความต้องการใช้งานในสภาพแวดล้อมที่แตกต่างกัน

2. การพัฒนาและติดตั้ง (Implement)

- ขั้นตอนนี้คือการนำแบบที่ออกแบบมาไปพัฒนาและติดตั้งจริง โดยการประกอบและเชื่อมต่ออุปกรณ์ต่าง ๆ เข้าด้วยกัน รวมถึงการเขียนโปรแกรมควบคุมปั๊มมอเตอร์และการตั้งค่าการตรวจจับของเซ็นเซอร์อินฟราเรด เพื่อให้ระบบสามารถทำงานได้อย่างมีประสิทธิภาพและตอบสนองต่อการใช้งานจริง

3. การทดสอบและผลลัพธ์ (Result)

- เมื่อพัฒนาระบบเสร็จสิ้น ระบบจะถูกทดสอบในสภาพแวดล้อมที่หลากหลายเพื่อประเมินประสิทธิภาพและความสามารถในการทำงาน ซึ่งจะรวมถึงการวัดความแม่นยำของการตรวจจับโดยเซ็นเซอร์อินฟราเรด การตอบสนองของปั๊มมอเตอร์ และการประเมินความสะดวกในการใช้งานของผู้ใช้

4. การสำรวจความพึงพอใจ (Survey Satisfaction)

- หลังจากการทดสอบเบื้องต้น จะมีการสำรวจความพึงพอใจของผู้ใช้ที่ทดลองใช้ระบบ โดยการเก็บข้อมูลจากการสอบถามความคิดเห็นและข้อเสนอแนะของผู้ใช้จริง ซึ่งข้อมูลเหล่านี้จะถูกนำมาวิเคราะห์เพื่อปรับปรุงและพัฒนาระบบให้ดียิ่งขึ้นในอนาคต

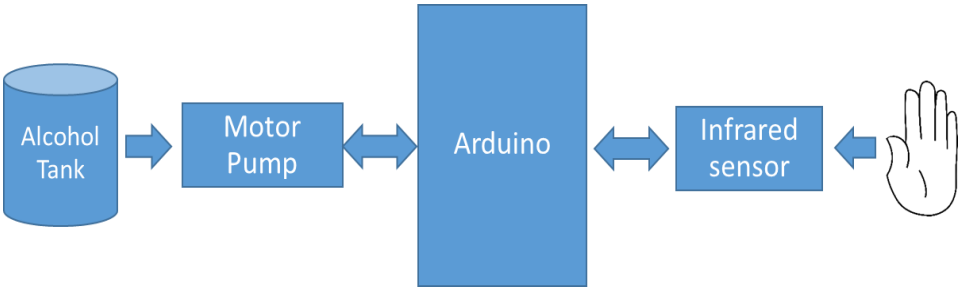
รูปที่ 7.1 วิธีการวิจัย

7.2.1 การออกแบบ

สามารถแบ่งออกเป็นสี่ขั้นตอน ดังนี้

1.การออกแบบ (Design)

สถาปัตยกรรมของระบบเครื่องจ่ายแอลกอฮอล์ล้างมืออัตโนมัติประกอบด้วยสามส่วนหลัก ดังแสดงในรูปที่ 7.2



รูปที่ 7.2 สถาปัตยกรรมของเครื่องจ่ายแอลกอฮอล์อัตโนมัติ

เซ็นเซอร์แบบอินฟราเรด

มือของผู้ใช้จะวางไว้ใต้เซ็นเซอร์อินฟราเรด เพื่อตรวจจับฝ่ามือและส่งข้อมูลไปยังบอร์ด Arduino

บอร์ด Arduino Nano3.0

เลือกใช้บอร์ด Arduino Nano3.0 เพราะต้องการบอร์ดที่มีขนาดเล็ก เพื่อรับสัญญาณจากเซ็นเซอร์ เมื่อมีเซ็นเซอร์ตรวจจับพบฝ่ามือในระยะการทำงาน จะส่งสัญญาณไฟเลี้ยงไปเปิดปั๊มมอเตอร์

ปั๊มมอเตอร์ (Pump Motor)

ปกติปั๊มมอเตอร์ จะไม่ทำงาน จนกว่าจะได้รับสัญญาณไฟเลี้ยงจากบอร์ด Arduino จึงจะทำงาน แล้วดูดและจ่ายแอลกอฮอล์ไหลออกมาจากหัวจ่ายได้

ขั้นตอนการทำงานของระบบ

เซ็นเซอร์อินฟราเรดของเครื่องจ่ายแอลกอฮอล์อัตโนมัติ ตรวจจับพลังงานอินฟราเรดที่มือปล่อยออกมาในรูปของความร้อน เมื่อมือวางไว้ในช่วงการทำงานของเซ็นเซอร์ พลังงานอินฟราเรดจะเปลี่ยนแปลงอย่างรวดเร็ว เพื่อกระตุ้นปั๊มทำงานเพื่อให้น้ำแอลกอฮอล์ไหลออกมาตามปริมาณที่ตั้งค่าไว้ในโปรแกรมของระบบควบคุม

7.2.2 อุปกรณ์ที่ใช้

เซ็นเซอร์อินฟราเรด

เซ็นเซอร์อินฟราเรด เป็นอุปกรณ์ที่ใช้ในการตรวจจับและวัดรังสีอินฟราเรด ซึ่งเป็นส่วนหนึ่งของคลื่นแม่เหล็กไฟฟ้าที่มีความยาวคลื่นยาวกว่าแสงที่ตามนุษย์มองเห็นได้ (โดยทั่วไปจะอยู่ในช่วงความยาวคลื่น 700 นาโนเมตร ถึง 1 มิลลิเมตร) เซ็นเซอร์นี้ทำงานโดยการปล่อยหรือรับคลื่นอินฟราเรดเพื่อการตรวจจับวัตถุหรือการวัดระยะทาง

ประเภทของเซ็นเซอร์อินฟราเรด

1. เซ็นเซอร์อินฟราเรดแบบแอคทีฟ (Active IR Sensors)

- การทำงาน ใช้การปล่อยคลื่นอินฟราเรดจากแหล่งกำเนิดและวัดการสะท้อนกลับของคลื่นนั้น

- การใช้งาน ใช้ในระบบป้องกันการบุกรุก, การตรวจจับการเคลื่อนไหว, และการวัดระยะทาง

- ตัวอย่าง รีโมทคอนโทรล, เซ็นเซอร์กันชนในหุ่นยนต์

2. เซ็นเซอร์อินฟราเรดแบบพาสซีฟ (Passive IR Sensors)

- การทำงานไม่ปล่อยคลื่นอินฟราเรดเอง แต่ตรวจจับรังสีอินฟราเรดที่ปล่อยออกมาจากวัตถุที่มีอุณหภูมิสูงกว่า

- การใช้งานในระบบตรวจจับการเคลื่อนไหว สำหรับระบบรักษาความปลอดภัย

- เซ็นเซอร์ตรวจจับความร้อนในอุปกรณ์ตรวจจับการเคลื่อนไหว

หลักการทำงานของเซ็นเซอร์อินฟราเรด

- การปล่อยและรับสัญญาณ เซ็นเซอร์จะปล่อยคลื่นอินฟราเรดไปยังวัตถุและรับสัญญาณสะท้อนกลับ การวัดเวลาหรือความเข้มของคลื่นที่สะท้อนกลับจะช่วยในการคำนวณระยะทางหรือการมีอยู่ของวัตถุ

- การตรวจจับความร้อน เซ็นเซอร์จะตรวจจับการเปลี่ยนแปลงของรังสีอินฟราเรดที่ปล่อยออกมาจากวัตถุ เมื่อมีวัตถุที่อุณหภูมิสูงเข้ามาในบริเวณเซ็นเซอร์ ความเข้มของรังสีอินฟราเรดที่ตรวจจับได้จะเพิ่มขึ้น

การใช้งานของเซ็นเซอร์อินฟราเรด

- การตรวจจับการเคลื่อนไหว ใช้ในระบบรักษาความปลอดภัยเพื่อตรวจจับการเคลื่อนไหวของบุคคล

- รีโมทคอนโทรล ใช้คลื่นอินฟราเรดในการส่งสัญญาณควบคุมอุปกรณ์อิเล็กทรอนิกส์

- ระบบนำทาง ใช้ในหุ่นยนต์และยานพาหนะเพื่อหลีกเลี่ยงการชน

- การวัดอุณหภูมิ ใช้ในอุปกรณ์วัดอุณหภูมิแบบไม่สัมผัส

ข้อดีและข้อเสียของเซ็นเซอร์อินฟราเรด

ข้อดี

- ความสามารถในการทำงานในที่มืด สามารถทำงานได้โดยไม่ต้องพึ่งพาแสงสว่างภายนอก

- ความแม่นยำในการตรวจจับ มีความไวต่อการเปลี่ยนแปลงของรังสีอินฟราเรด

- การใช้งานในหลากหลายสภาพแวดล้อม สามารถใช้งานได้ในสถานที่ที่มีฝุ่นหรือควัน

ข้อเสีย

- ระยะการตรวจจับจำกัด มีระยะการตรวจจับที่ค่อนข้างสั้นเมื่อเทียบกับเทคโนโลยีอื่น ๆ เช่น เรดาร์

- ความไวต่ออุณหภูมิ ประสิทธิภาพอาจลดลงเมื่อใช้งานในสภาพแวดล้อมที่มีความแตกต่างของอุณหภูมิสูง

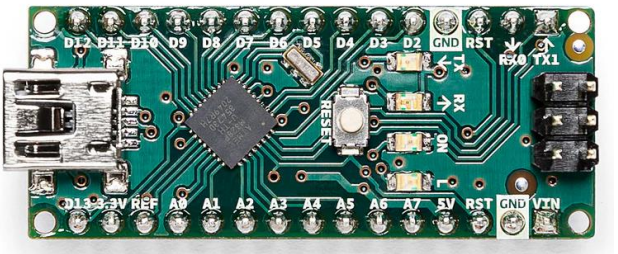
- การรบกวนจากแหล่งกำเนิดแสงอื่น คลื่นอินฟราเรดจากแหล่งอื่น ๆ เช่น แสงแดด อาจรบกวนการทำงานของเซ็นเซอร์

เซ็นเซอร์อินฟราเรด จะแสดงได้ดังรูปที่ 7.3 เป็นเซ็นเซอร์ที่มีความหลากหลายในการใช้งานและมีประโยชน์ในการตรวจจับและการวัดในหลายสถานการณ์ ตั้งแต่การใช้งานในชีวิตประจำวันไปจนถึงการใช้งานในอุตสาหกรรมและวิทยาศาสตร์

รูปที่ 7.3 เซ็นเซอร์อินฟราเรด

บอร์ด Arduino Nano 3.0

ในการวิจัยครั้งนี้ ได้ทำการเลือกใช้บอร์ด Arduino Nano 3.0 เนื่องจากต้องการใช้พื้นที่ให้น้อยที่สุดในกล่อง จึงเลือกใช้บอร์ด Arduino Nano3.0 เพราะมีขนาดเล็ก ทำงานได้ตามที่ออกแบบระบบได้ ซึ่งแสดงดังรูปที่ 7.4



https://store.arduino.cc/products/arduino-nano: Accessed: Jul. 2, 2024.

รูปที่ 7.4 บอร์ด Arduino Nano 3.0

รีเลย์ (Relay)

รีเลย์ (Relay) เป็นอุปกรณ์ไฟฟ้าที่ทำหน้าที่เป็นสวิตช์แม่เหล็กไฟฟ้า (Electromagnetic Switch) แสดงดังรูปที่ 7.5 ที่สามารถเปิดหรือปิดวงจรไฟฟ้าโดยการควบคุมด้วยสัญญาณไฟฟ้าที่เล็กกว่า ซึ่งการทำงานของรีเลย์สามารถแบ่งออกเป็นสองส่วนหลักๆ คือ ส่วนของวงจรควบคุม (Control Circuit) และส่วนของวงจรกำลัง (Power Circuit)

ส่วนประกอบและการทำงานของรีเลย์

1 ขดลวดแม่เหล็ก (Coil)

- ขดลวดนี้จะสร้างสนามแม่เหล็กเมื่อมีกระแสไฟฟ้าไหลผ่าน ทำหน้าที่เป็นส่วนของวงจรควบคุม (Control Circuit) เมื่อมีการกระตุ้นขดลวดนี้ รีเลย์จะทำงาน

- การควบคุมขดลวดนี้มักจะใช้สัญญาณไฟฟ้าขนาดเล็กที่มาจากวงจรควบคุม เช่น ไมโครคอนโทรลเลอร์หรือวงจรอิเล็กทรอนิกส์อื่นๆ

2 หน้าสัมผัส (Contacts)

- รีเลย์มีหน้าสัมผัสที่ทำหน้าที่เชื่อมต่อหรือแยกออกจากกันเพื่อเปิดหรือปิดวงจรไฟฟ้า ส่วนนี้จะทำหน้าที่เป็นวงจรกำลัง (Power Circuit)

- หน้าสัมผัสนี้สามารถเป็นแบบ Normally Open (NO) หรือ Normally Closed (NC) ซึ่งจะเปลี่ยนสถานะเมื่อขดลวดแม่เหล็กถูกกระตุ้น

3 อาร์เมเจอร์ (Armature)

- เป็นชิ้นส่วนที่เคลื่อนที่ได้และเชื่อมต่อกับหน้าสัมผัส อาร์เมเจอร์จะถูกดึงดูดหรือผลักเมื่อขดลวดแม่เหล็กถูกกระตุ้นหรือหยุดการกระตุ้น ทำให้หน้าสัมผัสเชื่อมต่อหรือแยกออกจากกัน

- เมื่อขดลวดได้รับพลังงาน สนามแม่เหล็กจะดึงอาร์เมเจอร์เพื่อเปลี่ยนสถานะของหน้าสัมผัสจากปกติไปยังสถานะทำงาน (หรือกลับกัน)

4 สปริง (Spring)

- ทำหน้าที่ดึงอาร์เมเจอร์กลับสู่ตำแหน่งเดิมเมื่อสนามแม่เหล็กหายไป ทำให้หน้าสัมผัสกลับสู่สถานะปกติ

การทำงานของรีเลย์

การทำงานของรีเลย์สามารถอธิบายได้เป็นขั้นตอนดังนี้

1 กระตุ้นขดลวดแม่เหล็ก เมื่อมีการจ่ายกระแสไฟฟ้าเข้าขดลวดแม่เหล็กของรีเลย์ สนามแม่เหล็กจะถูกสร้างขึ้นรอบขดลวดนี้

2 การเคลื่อนที่ของอาร์เมเจอร์ สนามแม่เหล็กที่สร้างขึ้นจะดึงหรือผลักอาร์เมเจอร์ ทำให้หน้าสัมผัสของรีเลย์เปลี่ยนสถานะ

- หากรีเลย์เป็นแบบ Normally Open (NO) หน้าสัมผัสจะปิดเชื่อมต่อวงจรไฟฟ้า

- หากรีเลย์เป็นแบบ Normally Closed (NC) หน้าสัมผัสจะเปิดแยกออกจากวงจรไฟฟ้า

3 การควบคุมวงจร การเปลี่ยนสถานะของหน้าสัมผัสทำให้สามารถควบคุมการเปิดหรือปิดวงจรที่ใช้พลังงานสูงกว่า เช่น การเปิดหรือปิดไฟฟ้าของมอเตอร์หรืออุปกรณ์ไฟฟ้าต่างๆ

4 การหยุดกระตุ้นขดลวด เมื่อการจ่ายกระแสไฟฟ้าไปยังขดลวดหยุดลง สนามแม่เหล็กจะหายไป และสปริงจะดึงอาร์เมเจอร์กลับสู่ตำแหน่งเดิม ทำให้หน้าสัมผัสกลับสู่สถานะปกติ

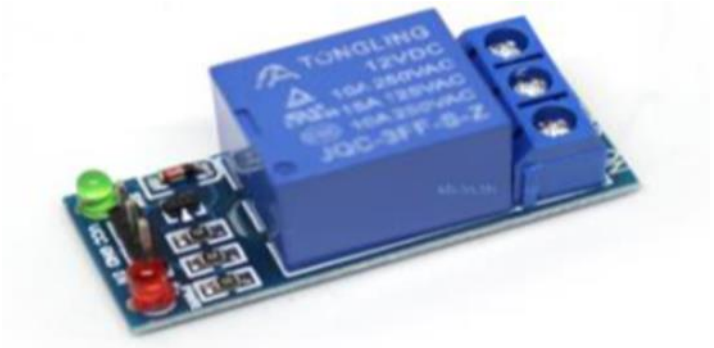
ประโยชน์ของการใช้รีเลย์

- การแยกวงจรควบคุมและวงจรกำลัง รีเลย์สามารถแยกวงจรที่ใช้สัญญาณไฟฟ้าขนาดเล็กเพื่อควบคุมวงจรที่ใช้พลังงานสูงได้

- ความปลอดภัย ช่วยป้องกันไม่ให้วงจรควบคุมถูกกระทบจากไฟฟ้ากระแสสูงในวงจรกำลัง

- การใช้งานที่หลากหลาย รีเลย์สามารถใช้ในแอปพลิเคชันต่างๆ เช่น การควบคุมแสงไฟ การควบคุมมอเตอร์ และการควบคุมระบบอัตโนมัติในอุตสาหกรรม

รีเลย์จึงเป็นอุปกรณ์ที่สำคัญในการควบคุมวงจรไฟฟ้า ช่วยให้สามารถเปิดหรือปิดวงจรที่ใช้พลังงานสูงได้โดยใช้สัญญาณควบคุมที่มีพลังงานต่ำกว่า เป็นส่วนประกอบที่พบได้ทั่วไปในระบบอิเล็กทรอนิกส์และการควบคุมอัตโนมัติ



รูปที่ 7.5 รีเลย์

ปั๊มมอเตอร์ (Pump Motor)

ปั๊มมอเตอร์ เป็นเครื่องจักรที่ใช้ในการขับเคลื่อนปั๊มสำหรับการเคลื่อนย้ายของเหลว (เช่น น้ำ, น้ำมัน, สารเคมี) จากที่หนึ่งไปยังอีกที่หนึ่ง มอเตอร์ทำหน้าที่เปลี่ยนพลังงานไฟฟ้าหรือเชื้อเพลิงเป็นพลังงานกล ซึ่งจะทำให้ปั๊มสามารถสร้างแรงดันและการไหลสำหรับการเคลื่อนย้ายของเหลวได้อย่างมีประสิทธิภาพ

ส่วนประกอบของปั๊มมอเตอร์

1. มอเตอร์ (Motor)

- มอเตอร์ไฟฟ้า (Electric Motor) เป็นชนิดที่ใช้พลังงานไฟฟ้าในการขับเคลื่อน เช่น มอเตอร์กระแสสลับ (AC) หรือกระแสตรง (DC)

- มอเตอร์เชื้อเพลิง (Fuel-powered Motor) ใช้เชื้อเพลิงเช่น น้ำมันดีเซลหรือเบนซินในการขับเคลื่อน เหมาะสำหรับการใช้งานในที่ที่ไม่มีไฟฟ้า

2. ปั๊ม (Pump)

- ปั๊มหมุนเวียน (Centrifugal Pump) ใช้ใบพัดในการเคลื่อนย้ายของเหลว

- ปั๊มลูกสูบ (Reciprocating Pump) ใช้ลูกสูบเพื่อสร้างแรงดันในการเคลื่อนย้ายของเหลว

- ปั๊มเฟือง (Gear Pump) ใช้เฟืองในการสร้างการไหลของของเหลว

หลักการทำงานของปั๊มมอเตอร์

1. การเปลี่ยนพลังงาน มอเตอร์จะเปลี่ยนพลังงานไฟฟ้าหรือเชื้อเพลิงเป็นพลังงานกลในการหมุนหรือการเคลื่อนที่

2. การสร้างแรงดัน พลังงานกลจากมอเตอร์จะถูกส่งไปยังปั๊ม ซึ่งจะสร้างแรงดันในการเคลื่อนย้ายของเหลว

3. การเคลื่อนที่ของของเหลว ของเหลวจะถูกดึงเข้ามาในปั๊มและถูกดันออกไปยังระบบท่อหรือถังเก็บตามแรงดันที่สร้างขึ้น

การใช้งานของปั๊มมอเตอร์

- อุตสาหกรรมน้ำและน้ำเสีย ใช้ในการเคลื่อนย้ายและบำบัดน้ำ

- ระบบระบายความร้อน ใช้ในระบบระบายความร้อนของเครื่องจักรและอุปกรณ์

- การเกษตร ใช้ในการสูบน้ำเพื่อการชลประทาน

- อุตสาหกรรมปิโตรเลียม ใช้ในการเคลื่อนย้ายและแปรรูปน้ำมันและก๊าซ

- การผลิตและแปรรูป ใช้ในการเคลื่อนย้ายสารเคมีและของเหลวในกระบวนการผลิต

ข้อดีและข้อเสียของปั๊มมอเตอร์

ข้อดี

- ความสามารถในการทำงานต่อเนื่อง ปั๊มมอเตอร์สามารถทำงานได้เป็นเวลานานโดยไม่หยุด

- ความหลากหลายในการใช้งาน มีการออกแบบที่หลากหลายสำหรับการใช้งานในสภาพแวดล้อมและความต้องการที่แตกต่างกัน

- ประสิทธิภาพสูง สามารถเคลื่อนย้ายของเหลวได้ปริมาณมากอย่างมีประสิทธิภาพ

ข้อเสีย

- ความต้องการการบำรุงรักษา ต้องการการบำรุงรักษาอย่างสม่ำเสมอเพื่อรักษาประสิทธิภาพ

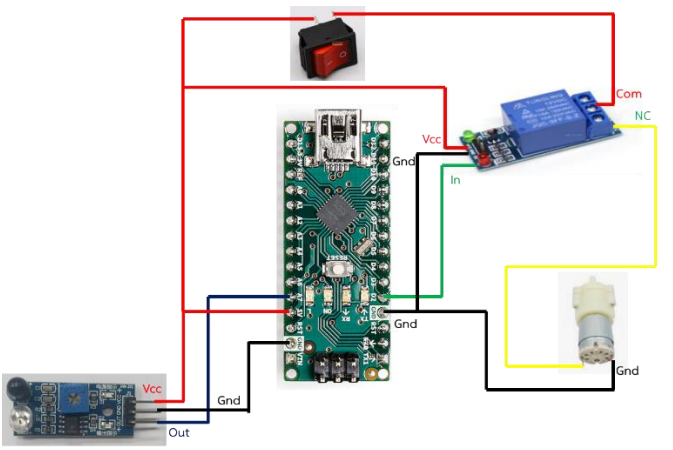
- ต้นทุนสูง อาจมีต้นทุนสูงในการติดตั้งและบำรุงรักษา

- การใช้พลังงาน มอเตอร์บางชนิดอาจใช้พลังงานมากและส่งผลต่อค่าใช้จ่ายในการดำเนินงาน

ปั๊มมอเตอร์ แสดงได้ดังรูปที่ 7.6 เป็นอุปกรณ์ที่สำคัญในการเคลื่อนย้ายของเหลวในหลายๆ ด้าน ทั้งในภาคอุตสาหกรรมและการใช้งานทั่วไปในชีวิตประจำวัน

รูปที่ 7.6 ปั๊มมอเตอร์

ให้ทำการต่ออุปกรณ์ ดังรูปที่ 7.7



รูปที่ 7.7 การต่อสายอุปกรณ์สำหรับเครื่องจ่ายแอลกอฮอล์อัตโนมัติ

พอประกอบเครื่องจ่ายแอลกอฮอล์อัตโนมัติ จะแสดงดังรูปที่ 7.8 ซึ่งเป็นรูปการติดตั้งอุปกรณ์ในเครื่องจ่ายแอลกอฮอล์อัตโนมัติ

7.2.3 การเขียนโปรแกรมสำหรับเครื่องจ่ายแอลกอฮอล์

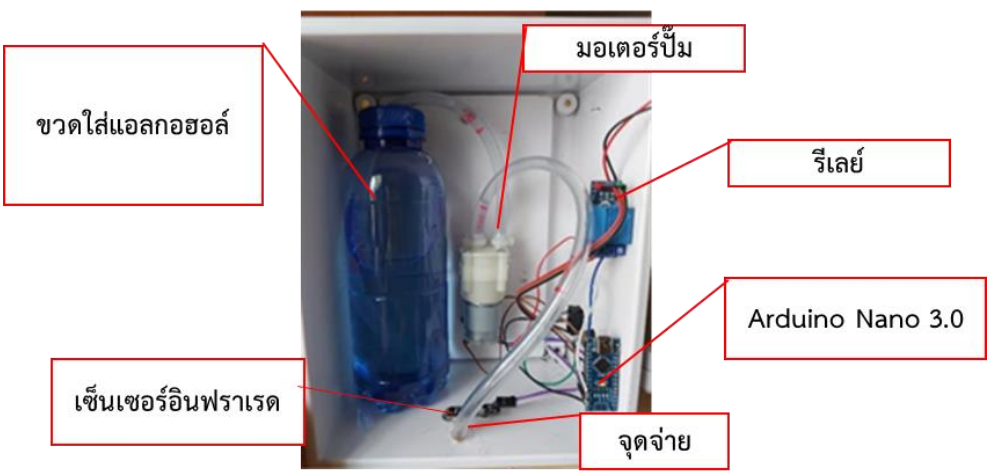
ให้ทำการเขียนโปรแกรมตามด้านล่างนี้

```
int ledPin = 2;  // กำหนดหมายเลขขาที่ 2 ของ LED
int sensor = A7; // กำหนดหมายเลขขาที่ 7 ของเซ็นเซอร์ที่อ่านค่าแบบอนาล็อก
int val = 0;     // ตัวแปรสำหรับเก็บค่าที่อ่านได้จากเซ็นเซอร์
int state;       // ตัวแปรสำหรับเก็บสถานะการทำงาน
void setup()
{
```

```
  pinMode(ledPin, OUTPUT); // กำหนดขา LED เป็นขาออก
  Serial.begin(9600);       // เริ่มการสื่อสารผ่าน Serial เพื่อการแสดงผลที่ความเร็ว 9600 บอด
  Serial.println("ArduinoAll TEST"); // แสดงข้อความบน Serial Monitor เมื่อเริ่มโปรแกรม
}
void loop()
{
  val = analogRead(sensor); // อ่านค่าจากขาเซ็นเซอร์ A7 (ค่าอนาล็อก)
  Serial.println(val);       // แสดงค่าที่อ่านได้จากเซ็นเซอร์ผ่าน Serial Monitor
  if (val > 500)  /* ตรวจสอบว่าค่าที่อ่านได้มากกว่า 500 หรือไม่ (ถ้ามีฝ่ามือกั้นที่เซ็นเซอร์อินฟราเรด ค่าจะ
มากกว่า 500 แต่ถ้าไม่มีฝ่ามือกั้นค่าจะน้อยกว่า 500 */
  {
    if (state == 1) // ถ้าสถานะการทำงานเป็น 1
    {
      digitalWrite(ledPin, HIGH); // เปิดปั๊มมอเตอร์ทำงานเพื่อดูดและจ่ายแอลกอฮอล์
      delay(100);                  // หน่วงเวลา 100 มิลลิวินาที
      digitalWrite(ledPin, LOW);  // ปิดปั๊มมอเตอร์ หยุดดูดและจ่ายแอลกอฮอล์
      state = 0;                   // เปลี่ยนสถานะเป็น 0
    }
  }
  else  // ถ้าค่าที่อ่านได้จากเซ็นเซอร์น้อยกว่าหรือเท่ากับ 500
  {
    digitalWrite(ledPin, LOW); // ปั๊มมอเตอร์ไม่ทำงาน
    state = 1;                  // ตั้งค่าสถานะเป็น 1
```

```
  }
  delay(100); // หน่วงเวลา 100 มิลลิวินาทีเพื่อให้วงจรทำงานได้เสถียร
}
```

แล้วให้ทำการอัปโหลดโปรแกรมลงบอร์ด Arduino Nano3.0



รูปที่ 7.8 ส่วนประกอบต่างๆ ในเครื่องจ่ายแอลกอฮอล์อัตโนมัติ

7.3 ผลการทดลอง

จากตารางที่ 7.1 ทำการทดสอบเครื่องจ่ายแอลกอฮอล์ที่ระยะห่างระหว่างมือกับเซ็นเซอร์ตั้งแต่ 0.5 ซ.ม. ถึง 2.5 ซม. เพื่อหาช่วงระยะที่เซ็นเซอร์สามารถตรวจจับได้อย่างมีประสิทธิภาพ ผลการทดสอบแสดงว่าเครื่องสามารถทำงานได้อย่างเต็มประสิทธิภาพที่ระยะ 0.5 ซ.ม. ถึง 2 ซ.ม. แต่ไม่สามารถทำงานได้ที่ระยะ 2.5 ซ.ม.

จากตารางที่ 7.2 ผลการสำรวจความพึงพอใจแสดงให้เห็นว่า คะแนนเฉลี่ยของผลการสำรวจความพึงพอใจ ได้ดังนี้

- ความสะดวกในการใช้งานและความสบาย 3.86
- ความสามารถในการลดการติดเชื้อ 3.96
- ความแม่นยำในการประมาณอัลกอฮอล์ในแต่ละครั้งที่ใช้ 3.93

ตารางที่ 7.1 ผลการวัดระยะการทำงานของเครื่องจ่ายแอลกอฮอล์อัตโนมัติ

ระยะห่างระหว่างฝ่ามือกับเซ็นเซอร์ / ครั้งที่	**0.5 ซ.ม.**	**1 ซ.ม.**	**1.5 ซ.ม.**	**2 ซ.ม.**	**2.5 ซ.ม.**
1	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ไม่ทำงาน
2	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ไม่ทำงาน
3	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ไม่ทำงาน
4	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ไม่ทำงาน
5	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ไม่ทำงาน
6	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ไม่ทำงาน
7	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ไม่ทำงาน
8	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ไม่ทำงาน
9	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ไม่ทำงาน
10	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ไม่ทำงาน
11	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ไม่ทำงาน
12	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ไม่ทำงาน
13	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ไม่ทำงาน
14	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ไม่ทำงาน
15	ทำงาน	ทำงาน	ทำงาน	ทำงาน	ไม่ทำงาน
เปอร์เซ็นเฉลี่ยในการทำงาน	**100%**	**100%**	**100%**	**100%**	**0%**

- ความชอบในด้านลักษณะภายนอกของอุปกรณ์ 3.24

- ความตอบสนองเมื่อเปรียบเทียบกับอุปกรณ์อื่น 3.93

- ความพึงพอใจทั่วไปต่ออุปกรณ์ 3.95

ตารางที่ 7.2 ความพึงพอใจของผู้ใช้จากการใช้เครื่องจ่ายแอลกอฮอล์อัตโนมัติที่มีเซ็นเซอร์อินฟราเรด โดยการการสำรวจนี้ได้จากผู้ใช้งานจำนวน 40 คน

ข้อที่	คำถาม	คะแนนเฉลี่ยที่ได้ (1คะแนนต่ำสุด: 5คะแนนสูงสุด)
1	ความง่ายต่อการใช้งานและสะดวกสบาย ของอุปกรณ์นี้ เมื่อเปรียบเทียบกับอุปกรณ์อื่น	3.86
2	ความสามารถลดการติดเชื้อได้ เมื่อเปรียบเทียบกับการต้องสัมผัสกับอุปกรณ์	3.96
3	การจ่ายปริมาณแอลกอฮอล์ที่แม่นยำในทุกครั้งที่ใช้งาน	3.93
4	คุณชอบลักษณะภายนอกของอุปกรณ์นี้มากน้อยเพียงใด เมื่อเปรียบเทียบกับอุปกรณ์อื่น	3.24
5	ความตอบสนองของอุปกรณ์นี้ เมื่อเปรียบเทียบกับอุปกรณ์อื่นที่เคยใช้มาก่อน	3.93
6	ความพึงพอใจภาพรวมต่ออุปกรณ์นี้ เมื่อเปรียบเทียบกับอุปกรณ์อื่นคือเท่าไหร่ครับ?	3.95

สรุปท้ายบท

การวิจัยนี้ได้พัฒนาและประเมินผลเครื่องจ่ายแอลกอฮอล์อัตโนมัติที่ใช้เซ็นเซอร์อินฟราเรดและบอร์ด Arduino ผลการทดลองแสดงให้เห็นว่าเครื่องนี้สามารถทำงานได้ดีที่ระยะระหว่าง 0.5 ซ.ม. ถึง 2 ซ.ม. และสามารถตั้งค่าเวลาการจ่ายแอลกอฮอล์ได้ตามต้องการ ผู้ใช้ส่วนใหญ่ให้ความเห็นว่าเครื่องนี้ใช้งานง่าย สะดวกสบาย และสามารถลดการติดเชื้อได้อย่างมีประสิทธิภาพ แม้ว่าในแง่ของรูปลักษณ์จะยังมีคะแนนต่ำกว่าด้านอื่น ๆ การพัฒนาเพิ่มเติมในอนาคตควรมุ่งเน้นไปที่การปรับปรุงรูปลักษณ์และความแม่นยำในการจ่ายแอลกอฮอล์ให้ดียิ่งขึ้น

คำถามท้ายบท

1.ให้อธิบายการทำงานของรีเลย์ พอสังเขป

2.ให้อธิบายการทำงานของเซ็นเซอร์อินฟราเรด

3.ให้อธิบายค่าสัญญาณอนาล็อกเมื่อมีฝ่ามือกั้นเซ็นเซอร์อินฟราเรดในเครื่องจ่ายแอลกอฮอล์ และกรณีไม่มีฝ่ามือกั้นเซ็นเซอร์อินฟราเรดค่าจะเป็นอย่างไร

4.ให้อธิบายความแตกต่างของเครื่องจ่ายแอลกอฮอล์ระหว่างบทนี้ และบทที่ 6

5.ให้ชี้แนะว่าควรพัฒนาเครื่องจ่ายแอลกอฮอล์นี้ ให้มีการพัฒนาขึ้นในด้านใดบ้าง พร้อมระบุเหตุผล

เอกสารอ้างอิง

1. N. Ferguson et al., "Impact of non-pharmaceutical interventions (NPIs) to reduce COVID19 mortality and healthcare demand," Report 9, 2020.

2. R. Guner, I. Hasanoglu, and F. Aktas, "COVID-19: Prevention and control measures in community," Turk J Med Sci, vol. 50(SI-1), pp. 571-577, 2020.

3. T. Jefferson et al., "Physical interventions to interrupt or reduce the spread of respiratory viruses: systematic review," BMJ, vol. 336(7635), pp. 77-80, 2008.

4. S.P. Luby et al., "Effect of handwashing on child health: a randomised controlled trial," The Lancet, vol. 366(9481), pp. 225-233, 2005.

5. X. Chen et al., "Hand Hygiene, Mask-Wearing Behaviors and Its Associated Factors during the COVID-19 Epidemic: A Cross-Sectional Study among Primary School Students in Wuhan, China," Int J Environ Res Public Health, vol. 17(8), 2020.

6. J.M. Boyce, "Measuring healthcare worker hand hygiene activity: current practices and emerging technologies," Infect Control Hosp Epidemiol, vol. 32(10), pp. 1016-1028, 2011.

7. A. Karn et al., "Design and Development of an Automated Monitored Hand Hygiene System to Curb Infection Spread in Institutional Settings during COVID-19 Pandemic," 2020.

8. A. Edgar-Tanzil, "Designing for a Pandemic: A Hands-Free Soap Dispeser," 2020.

9. A. Das et al., "Development of a Novel Design and Subsequent Fabrication of an Automated Touchless Hand Sanitizer Dispenser to Reduce the Spread of Contagious Diseases," Healthcare, vol. 9, pp. 1-17, 2021.

10. M.G. Rojo et al., "Non-contact temperature reader with sanitizer dispenser (NCTRSD)," Int. J. Sci. Res. Publ., vol. 10, pp. 583–593, 2020.

11. M.M. Srihari, "Self-activating sanitizer with battery imposed system for cleansing hands," in Proceedings of the 2nd International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2020, pp. 1102-1105.

12. Y. Eddy et al., "2019 Novel Coronavirus Disease (Covid-19): Smart contactless hand sanitizer-dispensing system using IoT based robotics technology," Rev. Argent. Clin. Psicol., vol. 29, pp. 215-220, 2020.

13. S. M. Rabeek et al. , " A reliable handwash detector for automated hand hygiene documentation and reminder system in hospitals," in The 15th International Conference on Biomedical Engineering, Springer: Cham, Switzerland, 2014, pp. 892-895.

14. R. Patra et al., "IoT-based computational frameworks in disease prediction and healthcare management: Strategies, challenges, and potential," in IoT in Healthcare and Ambient Assisted Living, Springer: Singapore, 2021, pp. 17-41.

15. W. Jongchanachavawat, B. Sungkongmueng, C. Inyasri, I. Roopkom, N. Mingmuang, P. Putthed, A. Srithammakan, B. Chinowan, P. Pathomphomma, P. Kumngern, S. Bunyarittikit, and P. Jongchanachavawat, "*Application of Ultrasonic Sensor and IR Sensor in Automatic Alcohol Hand Sanitizer*," Sensors and Materials, vol. 35, no. 4, 2023.

16. https://store.arduino.cc/products/arduino-nano: Accessed: Jul. 2, 2024.

ภาคผนวก

ก) โพรเจกต์เพื่อเป็นแบบฝึกหัดเสริมทักษะ

1. โพรเจกต์การตรวจจับการเคลื่อนไหว

   โดยใช้อุปกรณ์ดังนี้

   1.1 เซ็นเซอร์ตรวจจับความเคลื่อนไหว AM312 Micro PIR

   1.2 บอร์ด Arduino UNO

   1.3 ลำโพง (Buzzer)

   โจทย์ให้ตรวจจับ ความเคลื่อนไหวในระยะ 3-5 เมตร และเมื่อตรวจพบความเคลื่อนไหว ให้ระบบสั่งให้เสียงลำโพงดังขึ้น

2. โพรเจกต์การตรวจจับแสง

   โดยใช้อุปกรณ์ดังนี้

   2.1 โมดูลตรวจจับแสง(Light Sensor Module)

   2.2 บอร์ด Arduino UNO

   2.3 หลอดไฟที่ปรับค่าความสว่าง

   โจทย์ให้แสดงการตรวจจับแสง โดยการแสดงระดับความสว่าง 3 ระดับ พร้อมแสดงผลที่ Serial Monitor ข้อความ ระดับสว่างมาก สว่างปานกลาง และสว่างน้อย แสดงให้เห็นว่าสามารถตรวจจับแสงได้ถูกต้อง

3. โพรเจกต์การตั้งเวลาในการเปิดปิดไฟ

   อุปกรณ์ที่ใช้

   3.1 โมดูลนาฬิกา (DS3231 Module)

   3.2 บอร์ด Arduino UNO

   3.3 หลอดไฟแอลอีดี

   โจทย์ ต้องการให้ตั้งเวลาสำหรับการเปิดปิดไฟแอลอีดีได้ตามที่ต้องการ

ดรรชนี

ประวัติผู้เขียน

ผศ.ดร.วิโรจน์ จงชนะชววัฒน์

ประวัติการศึกษา

วศ.ด.(วิศวกรรมไฟฟ้า)	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
วศ.ม.(วิศวกรรมไฟฟ้า)	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
วศ.บ.(วิศวกรรมอิเล็กทรอนิกส์และการสื่อสาร)	มหาวิทยาลัยเอเชียอาคเนย์
MBA(Operation Management)	สถาบันบัณฑิตพัฒนบริหารศาสตร์
วท.บ.(โซลิตสเตทอิเล็กทรอนิกส์)	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประวัติการทำงาน

<u>ปัจจุบัน</u> ทำงานที่สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม

มหาวิทยาลัยราชภัฏเพชรบุรี

ทำงานมายาวนานกว่า 20 ปี ทางด้านระบบสารสนเทศและที่ปรึกษาในองค์กรเอกชน ในแวดวงอุตสาหกรรม ด้านอุตสาหกรรมอัญมณี ด้านอุตสาหกรรมอาหาร ด้านการศึกษา ด้านอุตสาหกรรมโรงแรมและรีสอร์ท ด้านอุตสาหกรรมยานยนต์ ด้านอุตสาหกรรมที่ปรึกษาทางธุรกิจ และด้านอุตสาหกรรมข้อมูลข่าวสารธุรกิจ