

	แผนการจัดการเรียนรู้	หน่วยที่ 1
	รหัสวิชา 20105-2005 ชื่อวิชา ไมโครคอนโทรลเลอร์	สอนครั้งที่ 1-2
	ชื่อหน่วยการเรียนรู้ที่ <i>งานใช้โปรแกรมไมโครคอนโทรลเลอร์เบื้องต้น</i>	ทฤษฎี 2 ชม. ปฏิบัติ 8 ชม.
ชื่อเรื่อง/งานใช้โปรแกรมไมโครคอนโทรลเลอร์เบื้องต้น		

1. ผลลัพธ์การเรียนรู้ระดับหน่วยการเรียนรู้

- 1.1 งานไมโครคอนโทรลเลอร์ Arduinoเบื้องต้น
- 1.2 งานโครงสร้างของไมโครคอนโทรลเลอร์
- 1.3 งานการติดตั้งโปรแกรม
- 1.4 งานทดสอบโปรแกรม
- 1.5 งานเขียนโปรแกรมและการอัปโหลดโปรแกรม

2. อ้างอิงมาตรฐาน/เชื่อมโยงกลุ่มอาชีพ

มาตรฐานฝีมือแรงงานแห่งชาติ กรมพัฒนาฝีมือแรงงาน หลักสูตรการฝึกยกระดับฝีมือ รหัสหลักสูตร 0920084190205 การประยุกต์ใช้ระบบสมองกลฝังตัวในโรงงานอุตสาหกรรม ระดับ 1

3. สมรรถนะประจำหน่วย

- 3.1 ติดตั้ง และเขียนโปรแกรมไมโครคอนโทรลเลอร์เบื้องต้น

4. จุดประสงค์เชิงพฤติกรรม

- 4.1 บอกความหมายของวงจรใช้งานไมโครคอนโทรลเลอร์
- 4.2 บอกหน้าที่ของโปรแกรมไมโครคอนโทรลเลอร์
- 4.3 บอกส่วนประกอบและการใช้งานของไมโครคอนโทรลเลอร์
- 4.4 ติดตั้งโปรแกรมเขียนโค้ดไมโครคอนโทรลเลอร์
- 4.5 มีกิจนิสัยที่ดีต่อการเรียนรู้และการปฏิบัติงานร่วมกับผู้อื่นด้วยความประณีต รอบคอบ

และปลอดภัย

5. สารการเรียนรู้

5.1 ไมโครคอนโทรลเลอร์ Arduino

Arduino (อาดูอีโน้ หรือ อาดูยโน้) มาจากภาษาอิตาลี เป็นบอร์ดไมโครคอนโทรลเลอร์ ตระกูล AVR ที่ถูกพัฒนาเป็นแบบโอเพ่นซอร์ส (Open Source) ซึ่งผู้ผลิตเปิดเผยข้อมูลทั้งฮาร์ดแวร์ (Hardware) และซอฟต์แวร์ (Software) บอร์ดไมโครคอนโทรลเลอร์ Arduino ถูกออกแบบขึ้นมาเพื่อให้ใช้งานได้ง่าย

5.2 โครงสร้างของไมโครคอนโทรลเลอร์ Atmega328

สำหรับไมโครคอนโทรลเลอร์ Atmega328 เป็นไอซีไมโครคอนโทรลเลอร์ของบริษัท Atmel มีโครงสร้างภายในเป็นแบบ RISC (Reduced instruction set computer) มีหน่วยความจำโปรแกรมภายในเป็นแบบแฟลช สามารถเขียน-ลบโปรแกรมใหม่ได้หลายครั้ง การโปรแกรมข้อมูลเป็นแบบ In-System programmable

5.3 สถาปัตยกรรมหลักของซีพียูตระกูล AVR

หน้าที่หลักของซีพียู คือ นำคำสั่งและข้อมูลที่เก็บไว้ในหน่วยความจำมาแปลความหมาย และกระทำตามเรียงกันไปทีละคำสั่ง ตามคำสั่งพื้นฐานของไมโครคอนโทรลเลอร์ ภายในซีพียูมีหน่วยคำนวณและตรรกะ หรือ Arithmetic & Logical Unit (ALU) เป็นหน่วยที่ทำหน้าที่คำนวณทางคณิตศาสตร์ ได้แก่ การบวก ลบ คูณ หาร และเปรียบเทียบทางตรรกะเพื่อทำการตัดสินใจ การทำงานของ ALU คือ รับข้อมูลจากหน่วยความจำ มาไว้ในที่เก็บชั่วคราวของ ALU หรือเรียกว่า รีจิสเตอร์ (register) เพื่อทำการคำนวณแล้วส่งผลลัพธ์กลับไปยังหน่วยความจำ

5.4 บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3

บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 ได้รับความนิยมอีกบอร์ดหนึ่ง เนื่องจากมีราคา ไม่แพง ซึ่งส่วนใหญ่โปรเจกต์และไลบรารีต่างๆ ที่พัฒนาขึ้นมาถูกอ้างอิงกับบอร์ดนี้เป็นหลัก เพราะเป็นขนาดที่เหมาะสมกับการเริ่มต้นการเรียนรู้ Arduino ซึ่งบอร์ด Arduino Uno ได้ถูกพัฒนาขึ้นมา ตั้งแต่ R2, R3 และมีรุ่นชิปไอซีเป็นแบบ SMD ในการเรียนรู้บอร์ดไมโครคอนโทรลเลอร์ Arduino เล่มนี้ใช้เป็นบอร์ดรุ่น Arduino Uno R3

6. กิจกรรมการเรียนรู้

6.1 ขั้นนำเข้าสู่บทเรียน

จัดให้นักเรียนศึกษาเริ่มต้นใช้งานไมโครคอนโทรลเลอร์ Arduino ในบทเรียน

1. ผู้สอนจัดเตรียมเอกสาร พร้อมกับแนะนำรายวิชา วิธีการให้คะแนนและวิธีการเรียนเรื่อง เริ่มต้นใช้งานไมโครคอนโทรลเลอร์ Arduino
2. ผู้สอนแจ้งจุดประสงค์การเรียนรู้ของหน่วยที่ 1 และขอให้ผู้เรียนร่วมกันทำกิจกรรมการเรียนรู้การสอน
3. ผู้สอนให้ผู้เรียนแสดงความรู้เกี่ยวกับ เรื่องเริ่มต้นใช้งานไมโครคอนโทรลเลอร์ Arduino

6.2 ขั้นให้ความรู้

1. ผู้สอนเปิด PowerPoint หน่วยที่ 1 เรื่อง เริ่มต้นใช้งานไมโครคอนโทรลเลอร์ Arduino
2. ผู้สอนอธิบายเนื้อหาในหน่วยที่ 1 เรื่อง เริ่มต้นใช้งานไมโครคอนโทรลเลอร์ Arduino

6.3 ขั้นประยุกต์ใช้

1. ผู้สอนให้ผู้เรียนทำใบงานที่ 1 เรื่อง เริ่มต้นใช้งานไมโครคอนโทรลเลอร์ Arduino
2. ผู้สอนให้ผู้เรียนทำแบบฝึกหัดหน่วยที่ 1 เริ่มต้นใช้งานไมโครคอนโทรลเลอร์ Arduino

6.4 ขั้นสรุปและประเมินผล

1. ผู้สอนและผู้เรียนร่วมกันสรุปเนื้อหาที่ได้เรียนให้มีความเข้าใจในทิศทางเดียวกัน
2. ผู้สอนให้ผู้เรียนศึกษาเพิ่มเติมนอกห้องเรียน จากเว็บไซต์เนื้อหาเกี่ยวกับ Arduino

7. สื่อและแหล่งการเรียนรู้

7.1 สื่อสิ่งพิมพ์

1. เอกสารประกอบการสอนวิชา ไมโครคอนโทรลเลอร์

2. ใบความรู้ที่ 1 เรื่อง เริ่มต้นใช้งานไมโครคอนโทรเลอร์ Arduino
3. แบบฝึกหัดหน่วยที่ 1 สรุปและประเมินผล
4. แบบประเมินผลงานตามใบงาน ใช้ประกอบการสอนชั้นประยุกต์ใช้
5. แบบประเมินพฤติกรรมการทำงาน ใช้ประกอบการสอนชั้นประยุกต์ใช้ ขึ้นสรุปและประเมินผล

7.2 สื่อโสตทัศน

1. เครื่องไมโครคอมพิวเตอร์
2. PowerPoint เรื่อง เริ่มต้นใช้งานไมโครคอนโทรเลอร์ Arduino
3. ห้องสมุดวิทยาลัยฯ

8. หลักฐานการเรียนรู้

8.1 หลักฐานความรู้

8.1.1 แบบวัดและประเมินผลความรู้

8.1.2 ใบรายการถาม ตอบ

8.2 หลักฐานการปฏิบัติงาน

8.2.1 ร้อยรอยการจดบันทึก

8.2.2 ใบงาน

8.2.3 การทดสอบภาคปฏิบัติ

9. การวัดและประเมินผล

9.1 เกณฑ์การปฏิบัติงาน

การสังเกต และพฤติกรรมระหว่างการเรียนรู้

9.2 วิธีการประเมิน

แบบทดสอบเฉพาะหน่วยและแบบวัดผลประเมินความรู้หลังเรียน

9.3 เครื่องมือประเมิน

ใบงาน แบบทดสอบเฉพาะหน่วย และแบบวัดผลประเมินความรู้หลังเรียน

10. บันทึกผลหลังการจัดการเรียนรู้

10.1 ข้อสรุปหลังการจัดการเรียนรู้

.....

.....

.....

10.2 ปัญหาที่พบ

.....

.....

.....

10.3 แนวทางแก้ปัญหา

	ใบความรู้	หน่วยที่ 1
	รหัสวิชา 20105-2005 ชื่อวิชา ไมโครคอนโทรลเลอร์	สอนครั้งที่ 1-2
	ชื่อหน่วยการเรียนรู้ งานใช้โปรแกรมไมโครคอนโทรลเลอร์เบื้องต้น	ทฤษฎี 2 ชม. ปฏิบัติ 8 ชม.
ชื่อเรื่อง งานใช้โปรแกรมไมโครคอนโทรลเลอร์เบื้องต้น		

1. ผลลัพธ์การเรียนรู้ระดับหน่วยการเรียนรู้

- 1.1 งานไมโครคอนโทรลเลอร์ Arduinoเบื้องต้น
- 1.2 งานโครงสร้างของไมโครคอนโทรลเลอร์
- 1.3 งานการติดตั้งโปรแกรม
- 1.4 งานทดสอบโปรแกรม
- 1.5 งานเขียนโปรแกรมและการอัปโหลดโปรแกรม

2. อ้างอิงมาตรฐาน/เชื่อมโยงกลุ่มอาชีพ

มาตรฐานฝีมือแรงงานแห่งชาติ กรมพัฒนาฝีมือแรงงาน หลักสูตรการฝึกยกระดับฝีมือ รหัสหลักสูตร 0920084190205 การประยุกต์ใช้ระบบสมองกลฝังตัวในโรงงานอุตสาหกรรม ระดับ 1

3. สมรรถนะประจำหน่วย

- 3.1 ติดตั้ง และเขียนโปรแกรมไมโครคอนโทรลเลอร์เบื้องต้น

4. จุดประสงค์เชิงพฤติกรรม

- 4.1 บอกความหมายของวงจรใช้งานไมโครคอนโทรลเลอร์
- 4.2 บอกหน้าที่ของโปรแกรมไมโครคอนโทรลเลอร์
- 4.3 บอกส่วนประกอบและการใช้งานของไมโครคอนโทรลเลอร์
- 4.4 ติดตั้งโปรแกรมเขียนโค้ดไมโครคอนโทรลเลอร์
- 4.5 มีกิจนิสัยที่ดีต่อการเรียนรู้และการปฏิบัติงานร่วมกับผู้อื่นด้วยความประณีต รอบคอบ

และปลอดภัย

5. สารการเรียนรู้

5.1 ไมโครคอนโทรลเลอร์ Arduino

Arduino (อาดูอินโน หรือ อาดูยโน) มาจากภาษาอิตาลี เป็นบอร์ดไมโครคอนโทรลเลอร์ ตระกูล AVR ที่ถูกพัฒนาเป็นแบบโอเพ่นซอร์ส (Open Source) ซึ่งผู้ผลิตเปิดเผยข้อมูลทั้งฮาร์ดแวร์ (Hardware) และซอฟต์แวร์ (Software) บอร์ดไมโครคอนโทรลเลอร์ Arduino ถูกออกแบบขึ้นมาเพื่อให้ใช้งานได้ง่าย

ฮาร์ดแวร์ (Hardware) เป็นบอร์ดอิเล็กทรอนิกส์ขนาดเล็กที่มีไมโครคอนโทรลเลอร์เป็นชิ้นส่วนหลัก ประกอบรวมกับ อุปกรณ์อิเล็กทรอนิกส์อื่น ๆ เพื่อให้ง่ายต่อการใช้งานหรือที่เรียกกันว่า “บอร์ด Arduino” โดยบอร์ด Arduino ก็มีหลายรุ่นให้เลือกใช้ตามความเหมาะสมของงาน โดยในแต่ละรุ่นอาจมีความแตกต่างกันในเรื่อง

ของขนาดของบอร์ด หรือสเปค เช่น จำนวนของขารับส่งสัญญาณ, แรงดันไฟที่ใช้, ประสิทธิภาพ ของ MCU เป็นต้น

ซอฟต์แวร์ (Software) - ภาษาที่ใช้เขียนโค้ดควบคุมบอร์ด Arduino เป็นภาษาสำหรับเขียนโปรแกรมควบคุมที่มีไวยากรณ์ แบบเดียวกับภาษา C/C++ - Arduino IDE เป็นเครื่องมือสำหรับเขียนโค้ดโปรแกรม การคอมไพล์โปรแกรม (การแปลงไฟล์ ภาษาซีให้เป็นภาษาเครื่อง) และอัปโหลดโปรแกรมลงบอร์ด

5.2 โครงสร้างของไมโครคอนโทรลเลอร์ Atmega328

สำหรับไมโครคอนโทรลเลอร์ Atmega328 เป็นไอซีไมโครคอนโทรลเลอร์ของบริษัท Atmel มีโครงสร้างภายในเป็นแบบ RISC (Reduced instruction set computer) มีหน่วยความจำโปรแกรมภายในเป็นแบบแฟลช สามารถเขียน-ลบโปรแกรมใหม่ได้หลายครั้ง การโปรแกรมข้อมูลเป็นแบบ In-System programmable

5.3 สถาปัตยกรรมหลักของซีพียูตระกูล AVR

หน้าที่หลักของซีพียู คือ นำคำสั่งและข้อมูลที่เก็บไว้ในหน่วยความจำมาแปลความหมาย และกระทำตามเรียงกันไปทีละคำสั่ง ตามคำสั่งพื้นฐานของไมโครคอนโทรลเลอร์ ภายในซีพียูมีหน่วยคำนวณและตรรกะ หรือ Arithmetic & Logical Unit (ALU) เป็นหน่วยที่ทำหน้าที่คำนวณทางคณิตศาสตร์ ได้แก่ การบวก ลบ คูณ หาร และเปรียบเทียบทางตรรกะเพื่อทำการตัดสินใจ การทำงานของ ALU คือ รับข้อมูลจากหน่วยความจำ มาไว้ในที่เก็บชั่วคราวของ ALU หรือเรียกว่า รีจิสเตอร์ (register) เพื่อทำการคำนวณแล้วส่งผลลัพธ์กลับไปยังหน่วยความจำ

5.4 บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3

บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 ได้รับความนิยมอีกบอร์ดหนึ่ง เนื่องจากมีราคา ไม่แพง ซึ่งส่วนใหญ่โปรเจกต์และไลบรารีต่างๆ ที่พัฒนาขึ้นมาถูกอ้างอิงกับบอร์ดนี้เป็นหลัก เพราะเป็นขนาดที่เหมาะสมกับการเริ่มต้นการเรียนรู้ Arduino ซึ่งบอร์ด Arduino Uno ได้ถูกพัฒนาขึ้นมา ตั้งแต่ R2, R3 และมีรุ่นชิปไอซีเป็นแบบ SMD ในการเรียนรู้บอร์ดไมโครคอนโทรลเลอร์ Arduino เล่มนี้ใช้เป็นบอร์ดรุ่น Arduino Uno R3

ตัวอย่างบอร์ด Arduino ที่นิยมนำมาใช้งาน



รูปที่ 1-3 บอร์ด Arduino UNO



รูปที่ 1-4 บอร์ด Arduino Nano

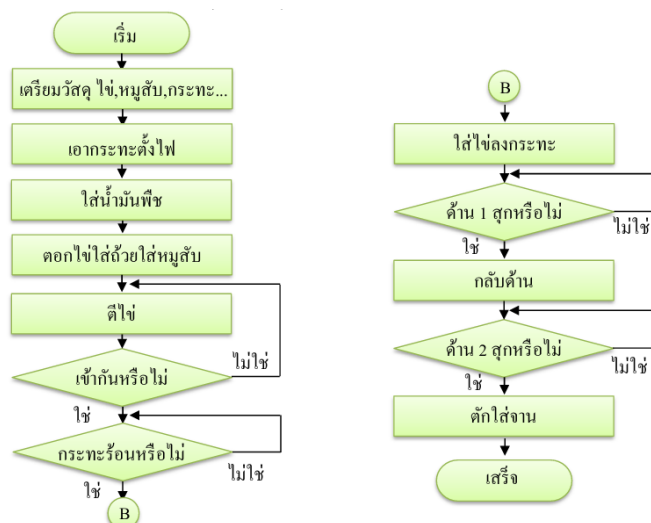


รูปที่ 1-5 บอร์ด Arduino Mega

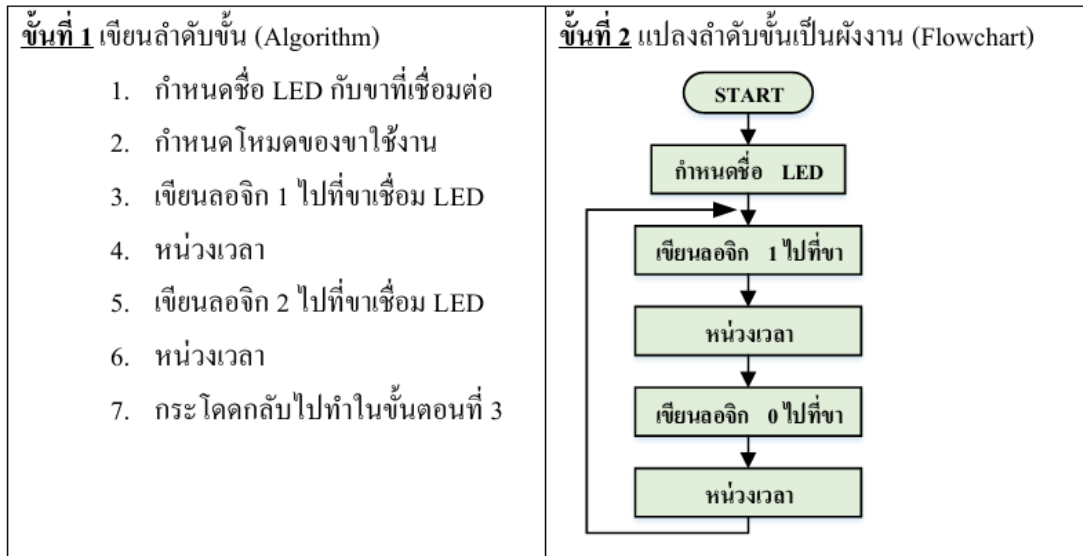
5.5 การเขียนโปรแกรมภาษาซีสำหรับไมโครคอนโทรลเลอร์

การใช้งานไมโครคอนโทรลเลอร์นอกจากจะต้องมีวงจรไมโครคอนโทรลเลอร์และมีส่วนของ วงจรเชื่อมต่อกับอุปกรณ์ภายนอกเพื่อใช้ในการควบคุมงานต่าง ๆ ตามต้องการที่เรียกว่าฮาร์ดแวร์ (Hardware) แล้วจำเป็นต้องมีชุดคำสั่งหรือโปรแกรมไว้สำหรับสั่งงานให้ไมโครคอนโทรลเลอร์ทำงาน ตามที่ต้องการซึ่งเรียกว่าซอฟต์แวร์ (Software)

ตัวอย่างการเขียนลำดับขั้นตอนวิธี และการแปลงเป็นผังงาน



ตัวอย่างการเขียนโปรแกรมไมโครคอนโทรลเลอร์



แปลงผังงานเป็นโปรแกรม

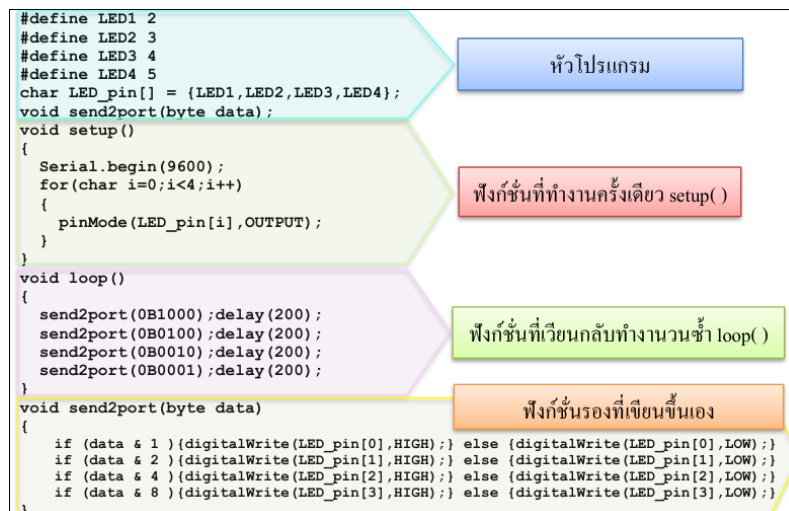
```

#define LED 13
void setup()
{
  pinMode(LED, OUTPUT);
}
void loop()
{
  digitalWrite(LED, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(LED, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
  
```

5.6 โครงสร้างของภาษาซีสำหรับ Arduino

โครงสร้างภาษาซีสำหรับ Arduino ถูกจัดใหม่ให้ง่ายต่อผู้ใช้งานเบื้องต้นซึ่งผู้ออกแบบได้จัดวาง ให้ผู้ใช้งานได้ใช้งานง่ายซึ่งโครงสร้างหลัก ๆ จะมีเพียง 2 ส่วนเท่านั้นคือ

1. setup เป็นส่วนที่เก็บฟังก์ชันที่ทำงานครั้งเดียว
2. loop เป็นส่วนที่เก็บฟังก์ชันที่เมื่อทำงานครบแล้วจะวนกลับมาทำซ้ำใหม่ตั้งแต่ต้น



5.7 ฟังก์ชันการดำเนินการแบบทางเลือก

โดยฟังก์ชันที่มึการทำงานแบบทางเลือก ในภาษาซีมีด้วยกัน 4 ฟังก์ชันคือ

1. ฟังก์ชัน if = (ทางเลือกเดียว)
2. ฟังก์ชัน if-else = (สองทางเลือก)
3. ฟังก์ชัน if-else if-else = (หลายทางเลือก)
4. ฟังก์ชัน switch = (หลายทางเลือก)

if (เงื่อนไขที่ตรวจสอบ)

ผังงาน	โค้ดโปรแกรม
<pre> graph TD Start(()) --> Cond{เงื่อนไข} Cond -- ใช่ --> TrueBox[ชุดฟังก์ชันเมื่อเงื่อนไขเป็นจริง] Cond -- ไม่ใช่ --> Exit(()) TrueBox --> Exit </pre>	<pre> if (conditional) { // put your code here // if conditional true } Example: if (value<50) { digitalWrite(13,HIGH); } </pre>

if (เงื่อนไขที่ตรวจสอบ) else

ผังงาน	โค้ดโปรแกรม
<pre> graph TD Start(()) --> Cond{เงื่อนไข} Cond -- ใช่ --> TrueBox[ชุดฟังก์ชันเมื่อเงื่อนไขเป็นจริง] Cond -- ไม่ใช่ --> FalseBox[ชุดฟังก์ชันเมื่อเงื่อนไขเป็นเท็จ] TrueBox --> Exit(()) FalseBox --> Exit </pre>	<pre> if (conditional) { // put main code here // if conditional true } else { //put main code here //if conditional false } </pre>

ฟังก์ชัน if-else แบบย่อ

โค้ดโปรแกรมฟังก์ชัน if-else แบบปกติ	โค้ดโปรแกรมฟังก์ชัน if-else แบบย่อ
<pre> if (SW1==LOW) { digitalWrite(LED,HIGH); } else{ digitalWrite(LED,LOW); } </pre>	<pre> digitalWrite(LED, (SW1==LOW) ? HIGH:LOW); </pre>

if (เงื่อนไขที่ตรวจสอบ) else if (เงื่อนไขที่ตรวจสอบ)

ผังงาน	โค้ดโปรแกรม
<pre> graph TD Start(()) --> Cond1{เงื่อนไขที่ 1} Cond1 -- ใช่ --> Func1[ชุดฟังก์ชันเมื่อ เงื่อนไข 1 เป็นจริง] Cond1 -- ไม่ใช่ --> Cond2{เงื่อนไขที่ 2} Cond2 -- ใช่ --> Func2[ชุดฟังก์ชันเมื่อ เงื่อนไข 2 เป็นจริง] Cond2 -- ไม่ใช่ --> Exit(()) Func1 --> Exit Func2 --> Exit </pre>	<pre> if (conditional1) { // put main code here // if conditional1 true } else if (conditional2) { //put main code here //if conditional2 true } </pre>

ฟังก์ชัน switch...case (หลายทางเลือก)

ผังงาน	โค้ดโปรแกรม
<pre> graph TD Start(()) --> Var[ตัวแปรที่จะตรวจสอบ] Var --> Cond1{เท่ากับค่าที่ 1} Cond1 -- จริง --> Func1[ชุดฟังก์ชันเมื่อ ค่าตัวแปรตรงกับค่าที่ 1] Cond1 -- เท็จ --> Cond2{เท่ากับค่าที่ 2} Cond2 -- จริง --> Func2[ชุดฟังก์ชันเมื่อ ค่าตัวแปรตรงกับค่าที่ 2] Cond2 -- เท็จ --> Exit[ชุดฟังก์ชันที่ตัวแปร ตรวจสอบไม่ตรงกับค่าใด] Func1 --> Exit Func2 --> Exit </pre>	<pre> switch (variable) { case 1: // put code here for case 1 break; case 2: // put code here for case 2 break; default: // put code here for default break; } </pre>

5.8 ฟังก์ชันการดำเนินการแบบวนซ้ำ

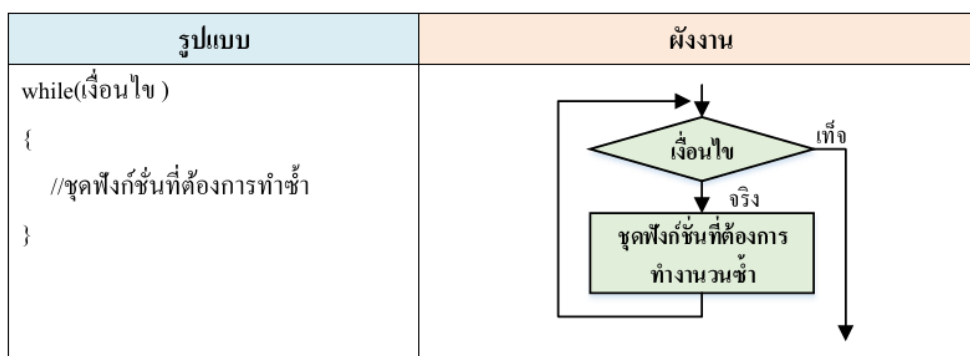
การเขียนโปรแกรมส่งงานไมโครคอนโทรลเลอร์ ต้องมีการทำงานแบบวนซ้ำหรือวนรอบ

1. ฟังก์ชัน for
2. ฟังก์ชัน while
3. ฟังก์ชัน while(1)
4. ฟังก์ชัน do-while

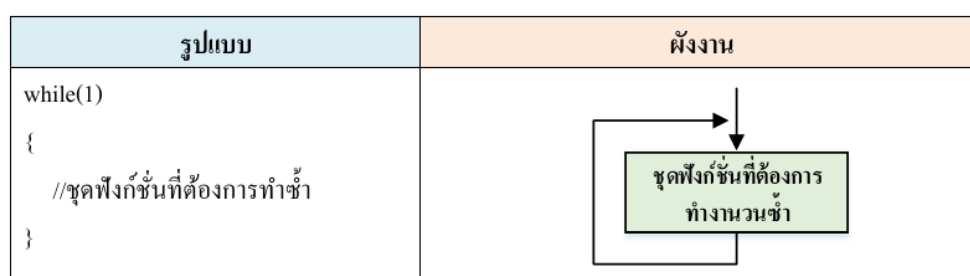
ฟังก์ชัน for

รูปแบบ	ผังงาน
<pre> for(ค่าเริ่มต้น;เงื่อนไข;เพิ่มหรือลดค่า) { //ชุดฟังก์ชันที่ต้องการทำซ้ำ } </pre>	<pre> graph TD Start(()) --> Init[ค่าตัวแปรนับรอบเริ่มต้น] Init --> Cond{เงื่อนไข} Cond -- จริง --> Func[ชุดฟังก์ชันที่ต้องการ ทำงานวนซ้ำ] Cond -- เท็จ --> Inc[เพิ่ม/ลดตัวแปรนับรอบ] Inc --> Cond </pre>

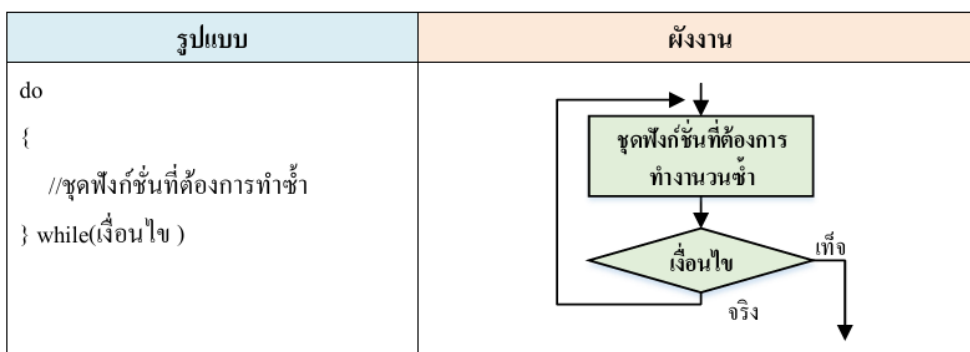
ฟังก์ชัน while



ฟังก์ชัน while(1)



ฟังก์ชัน do-while



5.9 การสร้างฟังก์ชันรองขึ้นใช้เอง

1. วางก่อนฟังก์ชันที่เรียกใช้งาน
2. วางหลังฟังก์ชันที่เรียกใช้งาน

วาก่อนฟังก์ชันที่เรียกใช้งาน	วากหลังฟังก์ชันที่เรียกใช้งาน
<pre> #define LED1 2 #define LED2 3 #define LED3 4 void pattern1(void) { digitalWrite(LED1,HIGH); digitalWrite(LED2,LOW); digitalWrite(LED3,LOW); } void pattern2(void) { digitalWrite(LED1,LOW); digitalWrite(LED2,HIGH); digitalWrite(LED3,LOW); } void pattern3(void) { digitalWrite(LED1,LOW); digitalWrite(LED2,LOW); digitalWrite(LED3,HIGH); } void setup() { pinMode(LED1,OUTPUT); pinMode(LED2,OUTPUT); pinMode(LED3,OUTPUT); } void loop() { pattern1(); delay(200); pattern2(); delay(200); pattern3(); delay(200); } </pre>	<pre> #define LED1 2 #define LED2 3 #define LED3 4 void pattern1(void); void pattern2(void); void pattern3(void); void setup() { pinMode(LED1,OUTPUT); pinMode(LED2,OUTPUT); pinMode(LED3,OUTPUT); } void loop() { pattern1(); delay(200); pattern2(); delay(200); pattern3(); delay(200); } void pattern1(void){ digitalWrite(LED1,HIGH); digitalWrite(LED2,LOW); digitalWrite(LED3,LOW); } void pattern2(void){ digitalWrite(LED1,LOW); digitalWrite(LED2,HIGH); digitalWrite(LED3,LOW); } void pattern3(void){ digitalWrite(LED1,LOW); digitalWrite(LED2,LOW); digitalWrite(LED3,HIGH); } </pre>

ฟังก์ชันรองที่เขียนขึ้นใช้เองมีด้วยกัน 4 ประเภทคือ

- ฟังก์ชันที่ไม่รับค่าและไม่ส่งคืนค่า
- ฟังก์ชันที่รับค่าแต่ไม่ส่งคืนค่า
- ฟังก์ชันที่ไม่รับค่าแต่ส่งคืนค่า
- ฟังก์ชันที่ค่าและส่งคืนค่า

ตัวอย่างฟังก์ชันที่ไม่รับค่าและไม่ส่งคืนค่า

```

1  #define LED1 2
2  #define LED2 3
3  #define LED3 4
4  void pattern1(void);
5  void pattern2(void);
6  void pattern3(void);
7  void setup()
8  {
9      pinMode(LED1,OUTPUT);
10     pinMode(LED2,OUTPUT);
11     pinMode(LED3,OUTPUT);
12 }
13 void loop()
14 {
15     pattern1(); delay(200);
16     pattern2(); delay(200);
17     pattern3(); delay(200);
18 }
19 void pattern1(void)
20 {
21     digitalWrite(LED1,HIGH);
22     digitalWrite(LED2,LOW);
23     digitalWrite(LED3,LOW);
24 }
25 void pattern2(void)
26 {
27     digitalWrite(LED1,LOW);
28     digitalWrite(LED2,HIGH);
29     digitalWrite(LED3,LOW);
30 }
31 void pattern3(void)
32 {
33     digitalWrite(LED1,LOW);
34     digitalWrite(LED2,LOW);
35     digitalWrite(LED3,HIGH);
36 }

```

ตัวอย่างฟังก์ชันที่รับค่าแต่ไม่ส่งคืนค่า

```

1  #define LED1 2
2  #define LED2 3
3  #define LED3 4
4  #define LED4 5
5  char LED_pin[] = {LED1,LED2,LED3,LED4};
6  void send2port(byte data);
7  void setup()
8  {
9      Serial.begin(9600);
10     for(char i=0;i<4;i++)
11     {
12         pinMode(LED_pin[i],OUTPUT);
13     }
14 }
15 void loop()
16 {
17     send2port(0B1000);delay(200);
18     send2port(0B0100);delay(200);
19     send2port(0B0010);delay(200);
20     send2port(0B0001);delay(200);
21 }
22 void send2port(byte data)
23 {
24     digitalWrite(LED_pin[0],(data & 1)? HIGH:LOW);
25     digitalWrite(LED_pin[1],(data & 2)? HIGH:LOW);
26     digitalWrite(LED_pin[2],(data & 4)? HIGH:LOW);
27     digitalWrite(LED_pin[3],(data & 8)? HIGH:LOW);
28 }

```

ตัวอย่างฟังก์ชันที่ไม่รับค่าแต่ส่งคืนค่า

```

1  #define LED 2
2  #define SW1 3
3  #define SW2 4
4  byte readSW(void);
5  void setup()
6  {
7      pinMode(LED,OUTPUT);
8      pinMode(SW1,INPUT_PULLUP);
9      pinMode(SW2,INPUT_PULLUP);
10 }

```

ตัวอย่างฟังก์ชันที่ค่าและส่งคืนค่า

```

1  #define NTC A5
2  double Thermistor(int RawADC);
3  void setup()
4  {
5      Serial.begin(9600);
6  }
7  void loop()
8  {
9      float Temp=Thermistor(analogRead(NTC));
10     Serial.print("Temperature is : ");
11     Serial.print(Temp);
12     Serial.println(" 'C");
13     delay(1000);
14 }
15 double Thermistor(int RawADC)
16 {
17     double Cal;
18     Cal = log(10000.0/((1024.0/RawADC-1)));
19     Cal = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 * Cal * Cal ))* Cal );
20     Cal = Cal - 273.15;           // Convert Kelvin to Celcius
21     return Cal;
22 }

```

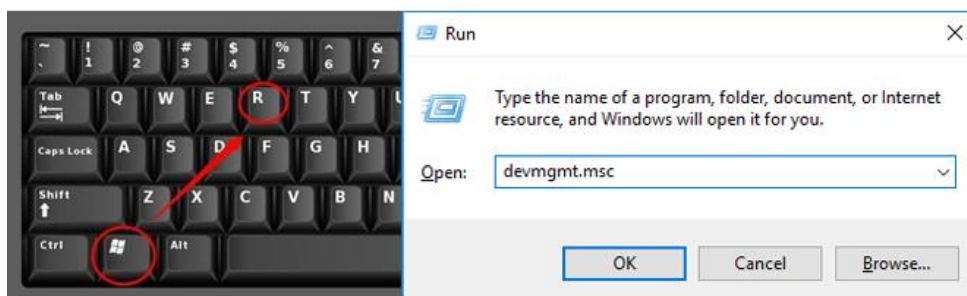
5.10 เครื่องมือช่วยงานไมโครคอนโทรลเลอร์ Arduino

การพัฒนาไม่ว่าจะเป็นการพัฒนาในเรื่องใดก็ตามหากมีเครื่องมือช่วยพัฒนางานที่มีคุณภาพจะ ทำให้งานนั้น ๆ สามารถพัฒนาไปได้เร็วและจะส่งผลให้ผู้พัฒนางานสามารถใช้ความสามารถได้เต็มที่สำหรับไมโครคอนโทรลเลอร์ในแพลตฟอร์ม Arduino มีเครื่องมือช่วยพัฒนางาน 3 ชนิด คือ

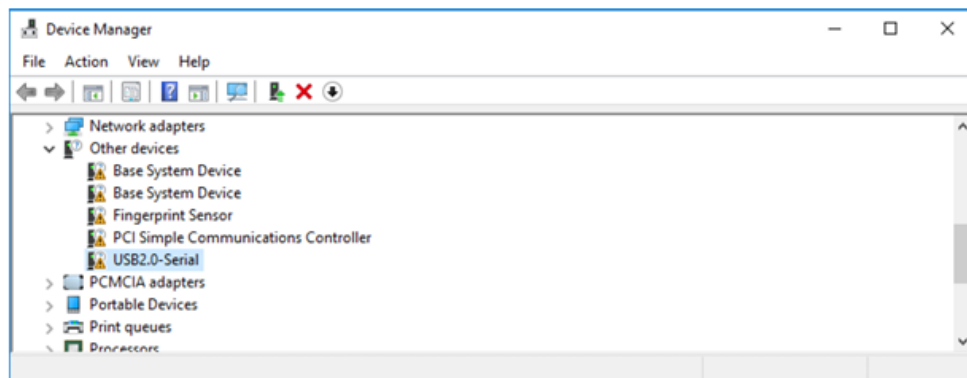
1. ไดรฟ์เวอร์ USB
2. เครื่องมือแปลงภาษาซีเป็นภาษาเครื่องและอัปโหลดไฟล์ลงชิพ (Arduino IDE)
3. เครื่องมือจำลองการทำงาน

1. ไดรฟ์เวอร์ USB

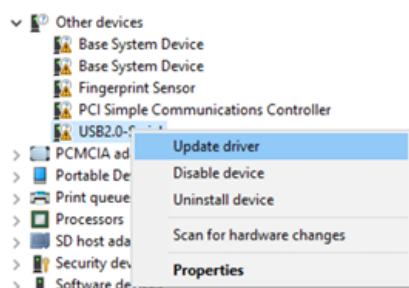
1.1 ขั้นตอนการติดตั้งไดรฟ์เวอร์ USB



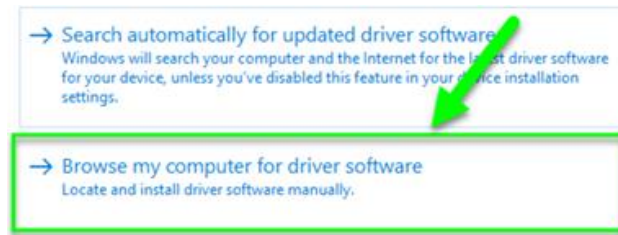
1.2 เชื่อมต่อบอร์ด Arduino เข้ากับคอมพิวเตอร์ สังเกตรายการพอร์ต USB ที่ปรากฏขึ้น



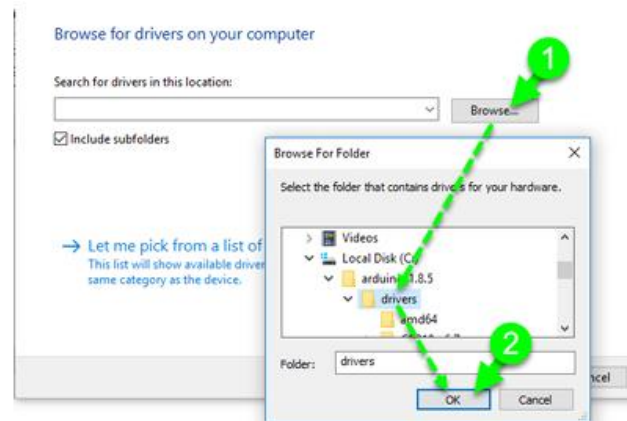
1.3 คลิกขวาที่ไอคอนเลือก Update driver



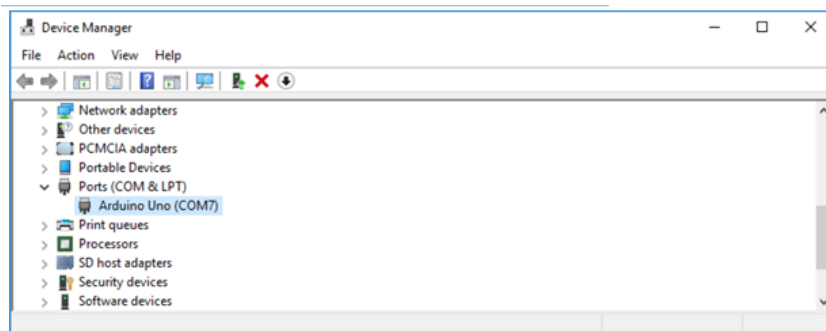
1.4 คลิก Browse my com... เพื่อหาไดรฟ์เวอร์ดังรูป



1.5 คลิกเลือกโฟลเดอร์ drivers ที่อยู่ภายในโฟลเดอร์ Arduino ดังรูป



1.6 ผลการติดตั้งไดรฟ์เวอร์ USB จากรูปเป็นบอร์ด Arduino UNO

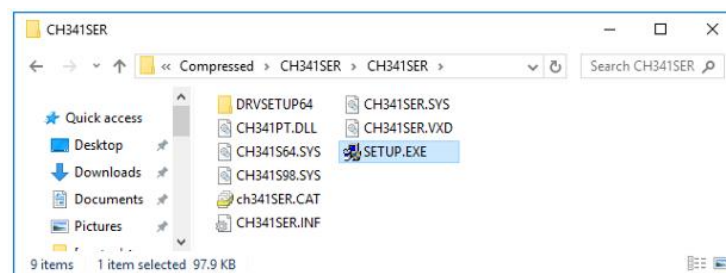


บอร์ด Arduino Nano 3.0 การดาวน์โหลดสามารถดาวน์โหลดไดรฟ์เวอร์ได้ที่

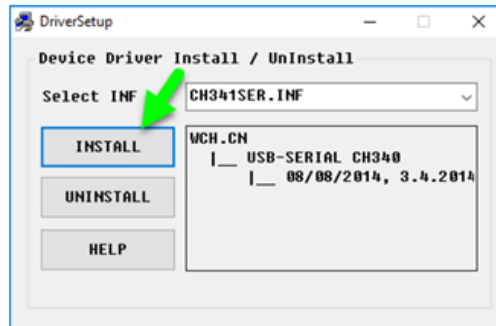
<http://www.winchiphead.com/download/CH341/CH341SER.ZIP>

การติดตั้งมีขั้นตอนดังนี้

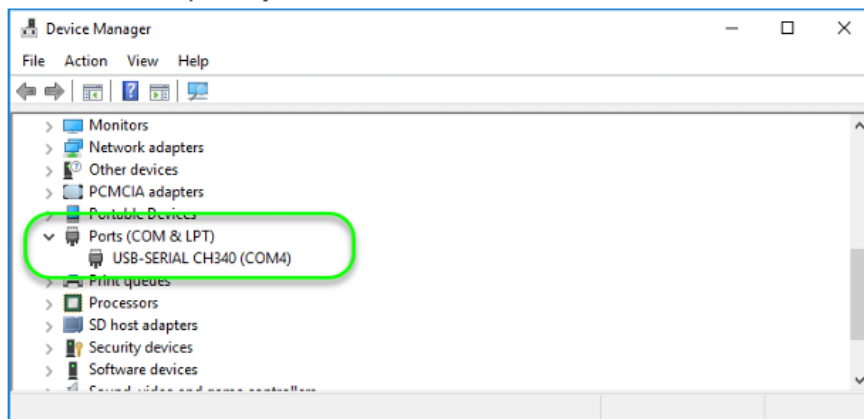
1. แยกไฟล์ zip ไดรฟ์เวอร์ที่ดาวน์โหลดจากลิงค์ที่กล่าวมาแล้วข้างต้น
2. เข้าไปในโฟลเดอร์ CH341SER แล้วดับเบิลคลิกที่ไฟล์ setup.exe



3. คลิกที่ปุ่ม install เพื่อดัดแปลงไดรฟ์เวอร์ USB ที่ใช้ชิพ CH340, CH341 ดังรูป



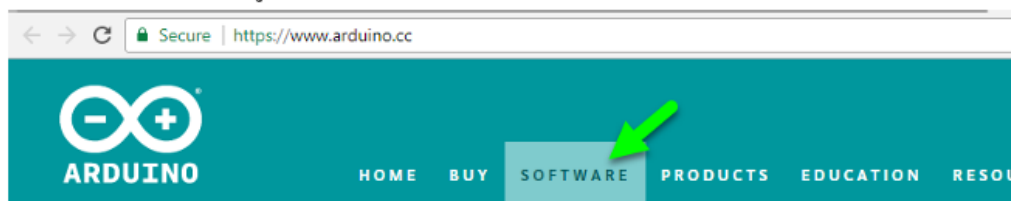
4. ตรวจสอบการทำงานของไดรฟ์เวอร์ด้วยการเชื่อมต่อบอร์ด Arduino เข้ากับคอมพิวเตอร์ด้วย สาย USB ทำการเปิด Device Manager



การติดตั้งโปรแกรม Arduino (IDE)

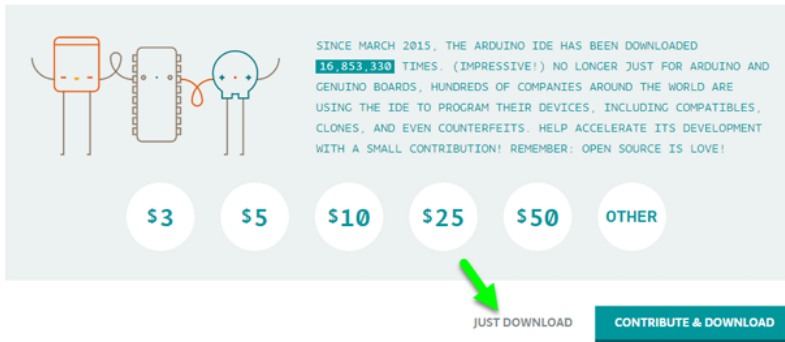
1.1 สามารถดาวน์โหลดซอฟต์แวร์โปรแกรม Arduino (IDE) โดยคลิกที่ Windows install จากเว็บไซต์

www.arduino.cc/en/Main/Software

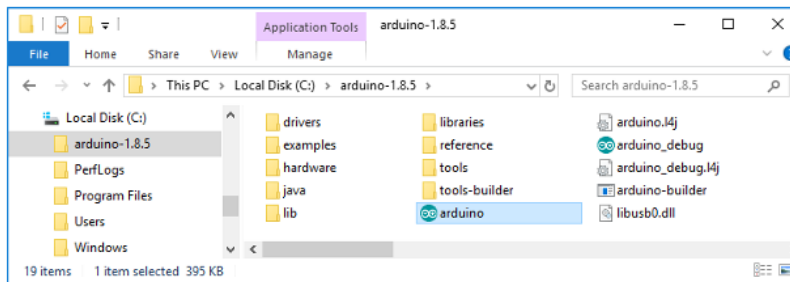


1.2 ทำการดับเบิลคลิกที่ไอคอน Arduino windows10



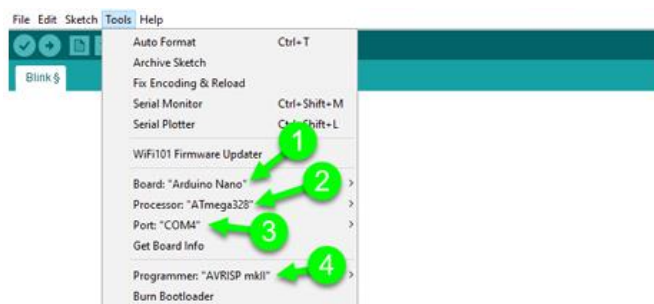


1.3 แยกไฟล์แล้วคัดลอกทั้งโฟลเดอร์มาไว้ที่ไดรฟ์ C ดังรูป

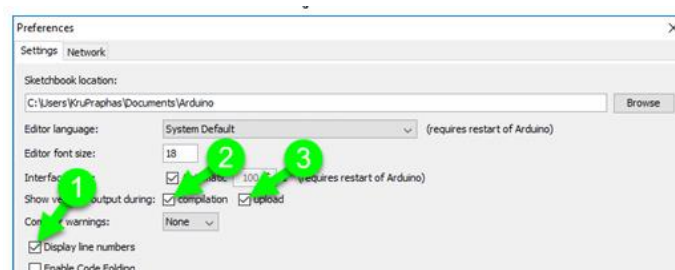


1.4 เมื่อติดตั้งเสร็จสิ้น สามารถคลิกเพื่อเข้าสู่โปรแกรม Arduino (IDE)

1.5 ชนิดของเครื่องโปรแกรม เลือก AVRISP mkII



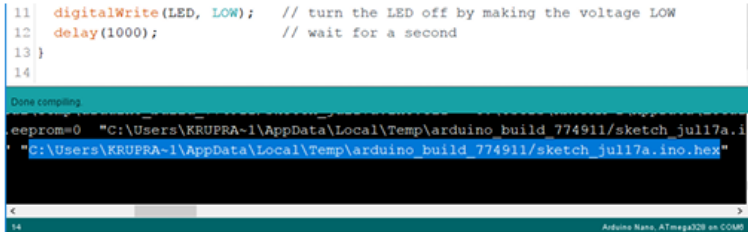
1.6 ตั้งค่าในโปรแกรม Arduino IDE ให้แสดงตำแหน่งของไฟล์ภาษาเครื่อง (HEX file) หลังจาก การคอมไพล์ผ่าน การตั้งค่าดังรูป



1.7 ในกรณีที่ต้องการคอมไพล์เพื่อแสดงผลของการเขียนโปรแกรมว่าถูกไวยกรณ์หรือไม่ หรือ ต้องการไฟล์ ภาษาเครื่อง (HEX file)



1.8 ในกรณีที่ต้องการไฟล์ภาษาเครื่องไปจำลองการทำงาน ให้คัดลอกตำแหน่งไฟล์ภาษาเครื่อง ในบริเวณหน้าต่างผลการคอมไพล์ (พื้นที่ดำ)

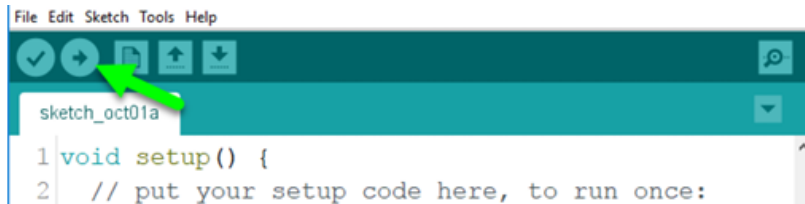


```

11 digitalWrite(LED, LOW); // turn the LED off by making the voltage LOW
12 delay(1000);           // wait for a second
13 }
14
Done compiling.
EEPROM=0 "C:\Users\KRUPRA-1\AppData\Local\Temp\arduino_build_774911/sketch_jul17a.i
"C:\Users\KRUPRA-1\AppData\Local\Temp\arduino_build_774911/sketch_jul17a.ino.hex"

```

1.9 ในกรณีที่ต้องการอัปโหลดโค้ดลงชิพของวงจรจริงทเชื่อมต่อไว้ให้คลิกที่ปุ่ม Upload ดังรูป



1.10 เมื่อเขียนโปรแกรมถูกต้อง โปรแกรมแสดงข้อความว่า Done compiling.

1.11 จากนั้นทำการอัปโหลดโปรแกรมที่คอมไพล์เรียบร้อยแล้ว โดยการกดที่ปุ่ม Upload

1.12 เมื่อทำการอัปโหลดเสร็จเรียบร้อยแล้ว โปรแกรมแสดงข้อความว่า Done uploading

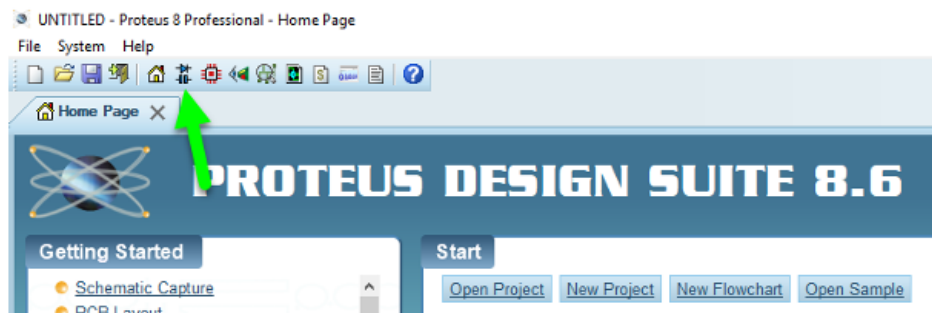
2. ส่วนประกอบและแถบเครื่องมือของโปรแกรม Arduino (IDE)

2.1 ส่วนประกอบของโปรแกรม Arduino (IDE)

2.2 แถบเครื่องมือเมนู มีแถบรายการของคำสั่งต่างๆ

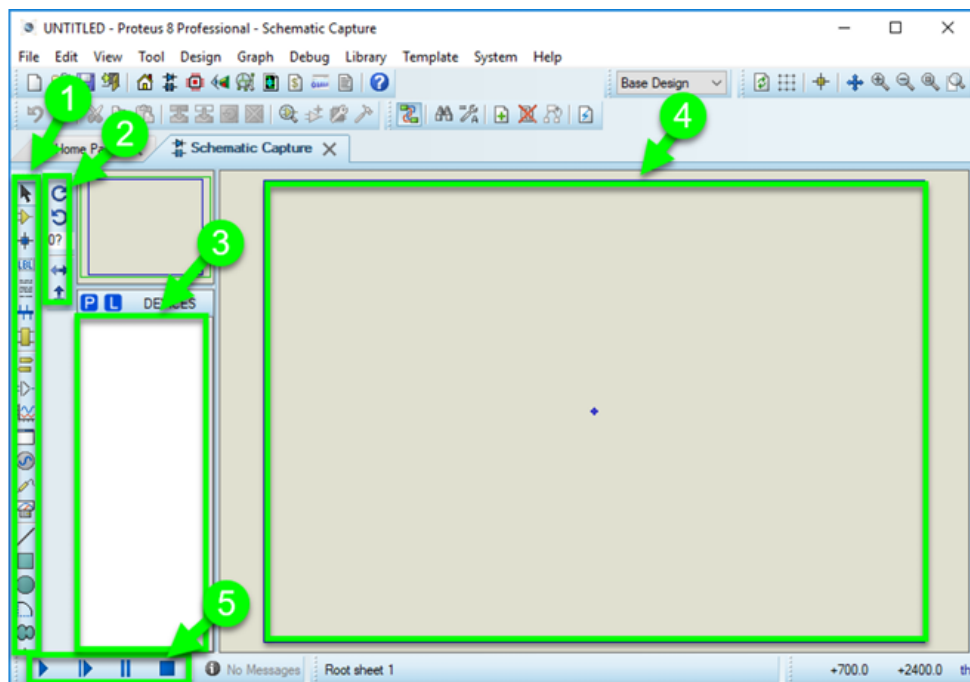
3. การจำลองการทำงานด้วยโปรแกรม Proteus

การจำลองการทำงานของวงจรอิเล็กทรอนิกส์ด้วยโปรแกรม Proteus เริ่มต้นจากการเรียกใช้ โปรแกรมซึ่งจะเป็นไอคอนอยู่หน้าเดสท็อป เรียกใช้งานในส่วน Schematic Capture

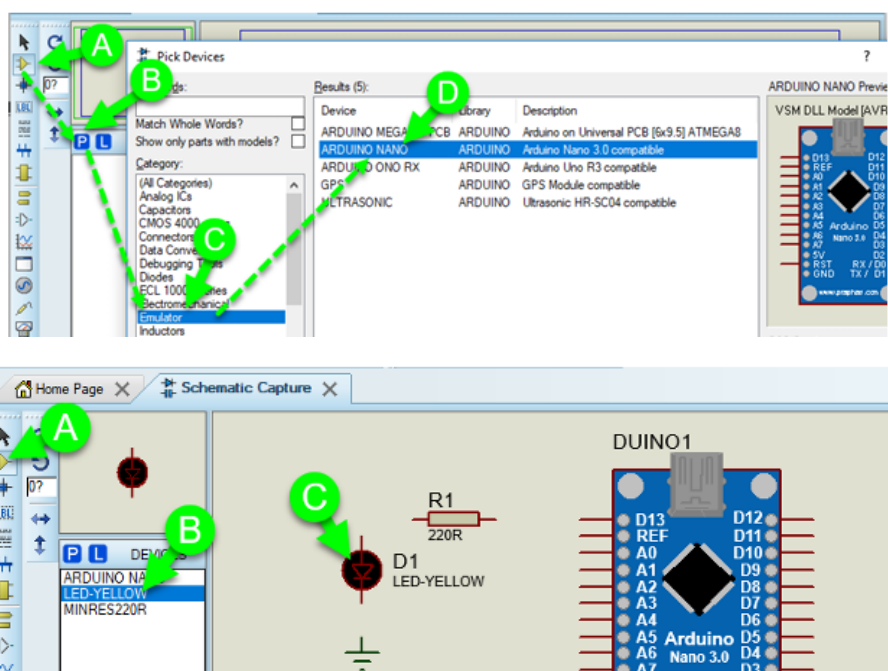


ส่วนประกอบต่าง ๆ ที่ใช้งานใน Schematic Capture ที่ใช้งานหลักมีดังนี้

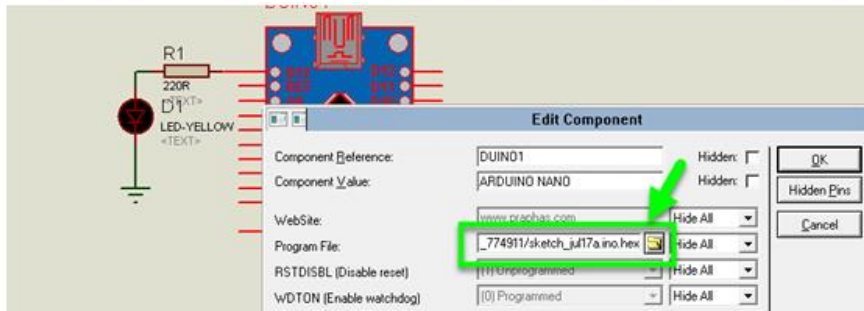
1. แถบเลือกโหมด
2. เครื่องมือหมุนหรือพลิกอุปกรณ์ที่เลือก
3. รายการอุปกรณ์ที่ถูกเลือกมาพร้อมนำไปใช้งาน
4. پنทวาดวงจรที่ใช้จำลองการทำงาน
5. เครื่องมือควบคุมการจำลองการทำงานของวงจร



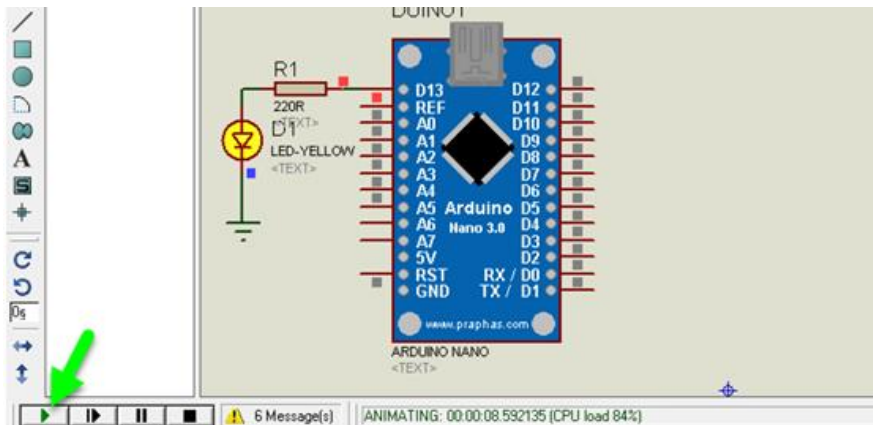
6. การเลือกอุปกรณ์



7. คัดลอกตำแหน่งไฟล์ภาษาเครื่องจากโปรแกรม Arduino IDE แล้วคลิกวางตำแหน่งไฟล์ ภาษาเครื่อง (Ctrl+V) ลงในช่อง Program File: โดยการดับเบิลคลิกที่ตัว Arduino



8. คลิกปุ่ม Run เพื่อเริ่มจำลองการทำงาน



6. แบบฝึกหัดท้ายหน่วย

ตอนที่ 1 ให้นักเรียนทำเครื่องหมายกากบาท (X) ลงบนคำตอบที่ถูกที่สุดเพียงคำตอบเดียว

1. Arduino ถูกผลิตขึ้นจากหน่วยงานใด

- ก. บริษัท intel
- ข. บริษัท Microchip
- ค. บริษัท Motorola
- ง. บริษัท Atmel

2. Arduino เป็นไมโครคอนโทรลเลอร์ ตระกูลใด

- ก. Z80 ข. AVR
- ค. MCS51 ง. PIC

3. ข้อใดไม่ใช่จุดเด่นของไมโครคอนโทรลเลอร์ Arduino

- ก. ราคาไม่แพง
- ข. เขียนโปรแกรมบนวินโดวส์ได้เท่านั้น
- ค. การใช้งานเป็นแบบโอเพ่นซอร์ส
- ง. มีรูปแบบคำสั่งพื้นฐาน ไม่ซับซ้อน

4. ข้อใดคือหน่วยความจำแบบแฟรช (Flash)

- ก. เขียนและลบโปรแกรมได้ครั้งเดียว

- ข. เขียนโปรแกรมได้หลายครั้งและลบโปรแกรมได้ครั้งเดียว
- ค. เขียนและลบโปรแกรมได้นับหมื่นครั้ง
- ง. เขียนโปรแกรมได้ครั้งเดียวและลบโปรแกรมได้หลายครั้ง

5. ไมโครคอนโทรลเลอร์ Atmega328 มีขนาดกี่บิต

- ก. 8 บิต ข. 10 บิต
- ค. 16 บิต ง. 24 บิต

6. ไมโครคอนโทรลเลอร์ Atmega328 มีขาทั้งหมดกี่ขา

- ก. 18 ขา ข. 24 ขา
- ค. 28 ขา ง. 32 ขา

7. ไมโครคอนโทรลเลอร์ Atmega328 มีหน่วยความจำโปรแกรมแบบ Flash ขนาดเท่าไร

- ก. 8 Kbyte ข. 16 Kbyte
- ค. 24 Kbyte ง. 32 Kbyte

8. หน่วย ALU ของไมโครคอนโทรลเลอร์ Atmega328 มีหน้าที่ใด

- ก. คำนวณทางคณิตศาสตร์และตรรกะ
- ข. เป็นการสร้างสัญญาณนาฬิกา
- ค. รับ-ส่งสัญญาณจากพอร์ต
- ง. หน่วยความจำภายใน

9. บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 มีพอร์ตดิจิทัลจำนวนเท่าใด

- ก. 10 ขา ข. 14 ขา
- ค. 16 ขา ง. 20 ขา

10. บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 มีพอร์ตอนาล็อกจำนวนเท่าใด

- ก. 4 ขา ข. 5 ขา
- ค. 6 ขา ง. 7 ขา

ตอนที่ 2 ให้นักเรียนตอบคำถามต่อไปนี้ให้ถูกต้อง

1. จงบอกข้อดีของบอร์ดไมโครคอนโทรลเลอร์ Arduino
2. จงบอกคุณสมบัติเบื้องต้นของไอซีเบอร์ Atmega328

7. เอกสารอ้างอิง

บรรณานุกรม ภาษาไทย ไมโครคอนโทรลเลอร์ : ศูนย์ส่งเสริมอาชีพ, 2567.

(สาโรช กล่อมอญ, 2562)

8. ภาคผนวก (เฉลยแบบฝึกหัด เฉลยแบบทดสอบ ฯ)

ตอนที่ 1 ให้นักเรียนทำเครื่องหมายกากบาท (X) ลงบนคำตอบที่ถูกต้องเพียงคำตอบเดียว

1. Arduino ถูกผลิตขึ้นจากหน่วยงานใด
 - ก. บริษัท intel
 - ข. บริษัท Microchip
 - ค. บริษัท Motorola
 - ง. บริษัท Atmel
2. Arduino เป็นไมโครคอนโทรลเลอร์ ตระกูลใด
 - ก. Z80
 - ข. AVR
 - ค. MCS51
 - ง. PIC
3. ข้อใดไม่ใช่จุดเด่นของไมโครคอนโทรลเลอร์ Arduino
 - ก. ราคาไม่แพง
 - ข. เขียนโปรแกรมบนวินโดวส์ได้เท่านั้น
 - ค. การใช้งานเป็นแบบโอเพ่นซอร์ส
 - ง. มีรูปแบบคำสั่งพื้นฐาน ไม่ซับซ้อน
4. ข้อใดคือหน่วยความจำแบบแฟรช (Flash)
 - ก. เขียนและลบโปรแกรมได้ครั้งเดียว
 - ข. เขียนโปรแกรมได้หลายครั้งและลบโปรแกรมได้ครั้งเดียว
 - ค. เขียนและลบโปรแกรมได้นับหมื่นครั้ง
 - ง. เขียนโปรแกรมได้ครั้งเดียวและลบโปรแกรมได้หลายครั้ง
5. ไมโครคอนโทรลเลอร์ Atmega328 มีขนาดกี่บิต
 - ก. 8 บิต
 - ข. 10 บิต
 - ค. 16 บิต
 - ง. 24 บิต
6. ไมโครคอนโทรลเลอร์ Atmega328 มีขาทั้งหมดกี่ขา
 - ก. 18 ขา
 - ข. 24 ขา
 - ค. 28 ขา
 - ง. 32 ขา
7. ไมโครคอนโทรลเลอร์ Atmega328 มีหน่วยความจำโปรแกรมแบบ Flash ขนาดเท่าไร
 - ก. 8 Kbyte
 - ข. 16 Kbyte
 - ค. 24 Kbyte
 - ง. 32 Kbyte
8. หน่วย ALU ของไมโครคอนโทรลเลอร์ Atmega328 มีหน้าที่ใด
 - ก. คำนวณทางคณิตศาสตร์และตรรกะ
 - ข. เป็นการสร้างสัญญาณนาฬิกา
 - ค. รับ-ส่งสัญญาณจากพอร์ต
 - ง. หน่วยความจำภายใน

9. บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 มีพอร์ตดิจิตอลจำนวนเท่าใด

ก. 10 ขา **ข. 14 ขา**

ค. 16 ขา ง. 20 ขา

10. บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 มีพอร์ตอนาล็อกจำนวนเท่าใด

ก. 4 ขา ข. 5 ขา

ค. 6 ขา ง. 7 ขา

ตอนที่ 2 ให้นักเรียนตอบคำถามต่อไปนี้ให้ถูกต้อง

1. จงบอกข้อดีของบอร์ดไมโครคอนโทรลเลอร์ Arduino

1. มีรูปแบบคำสั่งพื้นฐาน ไม่ซับซ้อนเหมาะสำหรับผู้เริ่มต้นใช้งาน และง่ายต่อการพัฒนาโปรแกรม
2. การใช้งานเป็นแบบโอเพ่นซอร์ส (Open Source) สามารถนำบอร์ดไปต่อยอดใช้งานได้หลายด้าน
3. สามารถพัฒนาโปรแกรมบนระบบปฏิบัติการต่างๆ ได้เช่น Windows, Mac OS X หรือ Linux
4. บอร์ดไมโครคอนโทรลเลอร์ Arduino มีราคาไม่แพง

2. จงบอกคุณสมบัติเบื้องต้นของไอซีเบอร์ Atmega328

1. เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิตแบบ RISC กำลังงานต่ำ
2. หน่วยความจำโปรแกรมแบบแฟลช 32 กิโลไบต์ สามารถเขียนและลบโปรแกรมในวงจร ได้ 10,000 รอบ เก็บรักษาข้อมูลได้นาน 20 ปี
3. หน่วยความจำข้อมูลชั่วคราวแบบ SRAM 2 กิโลไบต์ และหน่วยความจำข้อมูลถาวรแบบ EEPROM 1 กิโลไบต์
4. บอร์ด USB บุตรโหนดเตอร์ มาจากผู้ผลิต จึงโปรแกรมหน่วยความจำผ่านพอร์ต USB ได้ โดยไม่ต้องใช้เครื่องโปรแกรมภายนอก
5. พอร์ตอินพุตเอาต์พุตแบบโปรแกรมได้ รวม 23 ขา
6. ความถี่สัญญาณนาฬิกาจากภายนอกสูงสุด 20 MHz
7. มีโมดูลแปลงสัญญาณอนาล็อกเป็นดิจิตอล ความละเอียด 10 บิตจำนวน 6 ช่อง
8. มีโมดูลไทมเมอร์/เคาน์เตอร์ ขนาด 16 บิตจำนวน 1 ช่อง, ขนาด 8 บิตจำนวน 2 ช่อง
9. มีโมดูลกำเนิดสัญญาณ PWM จำนวน 6 ช่อง
10. มีวงจรตรวจจับไฟเลี้ยงต่ำกว่าที่กำหนดหรือบรวเอาต์แบบโปรแกรมได้
11. มีโมดูลสื่อสารข้อมูลผ่านบัส SPI และ I²C
12. มีโมดูลอินเทอร์รัพท์ภายนอก 2 ช่องและภายใน 26 ช่อง
13. มีโมดูลสื่อสารข้อมูลอนุกรม USART
14. ใช้ไฟเลี้ยงมีค่าแรงดันตั้งแต่ 4.5 - 5.5V ถ้าเลือกใช้สัญญาณนาฬิกาที่ความถี่ 0 - 20MHz
15. ค่าอุณหภูมิใช้งานตั้งแต่ -40 ถึง +85 °C

	ใบงาน	หน่วยที่ 1
	รหัสวิชา 20105-2005 ชื่อวิชา ไมโครคอนโทรลเลอร์	สอนครั้งที่ 1-2
	ชื่อหน่วยการเรียนรู้ <i>งานใช้โปรแกรมไมโครคอนโทรลเลอร์เบื้องต้น</i>	ทฤษฎี 2 ชม. ปฏิบัติ 8 ชม.
ชื่อเรื่อง <i>งานใช้โปรแกรมไมโครคอนโทรลเลอร์เบื้องต้น</i>		

1. ผลลัพธ์การเรียนรู้ระดับหน่วยการเรียนรู้

- 1.1 งานไมโครคอนโทรลเลอร์ Arduinoเบื้องต้น
- 1.2 งานโครงสร้างของไมโครคอนโทรลเลอร์
- 1.3 งานการติดตั้งโปรแกรม
- 1.4 งานทดสอบโปรแกรม
- 1.5 งานเขียนโปรแกรมและการอัปโหลดโปรแกรม

2. อ้างอิงมาตรฐาน/เชื่อมโยงกลุ่มอาชีพ

มาตรฐานฝีมือแรงงานแห่งชาติ กรมพัฒนาฝีมือแรงงาน หลักสูตรการฝึกยกระดับฝีมือ รหัสหลักสูตร 0920084190205 การประยุกต์ใช้ระบบสมองกลฝังตัวในโรงงานอุตสาหกรรม ระดับ 1

3. สมรรถนะประจำหน่วย

- 3.1 ติดตั้ง และเขียนโปรแกรมไมโครคอนโทรลเลอร์เบื้องต้น

4. จุดประสงค์เชิงพฤติกรรม

- 4.1 บอกความหมายของวงจรใช้งานไมโครคอนโทรลเลอร์
- 4.2 บอกหน้าที่ของโปรแกรมไมโครคอนโทรลเลอร์
- 4.3 บอกส่วนประกอบและการใช้งานของไมโครคอนโทรลเลอร์
- 4.4 ติดตั้งโปรแกรมเขียนโค้ดไมโครคอนโทรลเลอร์
- 4.5 มีกิจนิสัยที่ดีต่อการเรียนรู้และการปฏิบัติงานร่วมกับผู้อื่นด้วยความประณีต รอบคอบ

และปลอดภัย

5. เครื่องมือและอุปกรณ์

- | | |
|------------------------------------|-----------|
| 1. บอร์ดทดลอง Breadboard 830 Point | 1 แผง |
| 2. อุปกรณ์อิเล็กทรอนิกส์ประกอบด้วย | |
| 2.1 ตัวต้านทาน 220 ohm | 1 ตัว |
| 2.2 LED 3mm | 1 ตัว |
| 3. บอร์ด Arduino Nano 3.0 | 1 ตัว |
| 4. สายเชื่อมต่อ USB (Mini USB) | 1 ชุด |
| 5. คอมพิวเตอร์ | 1 เครื่อง |

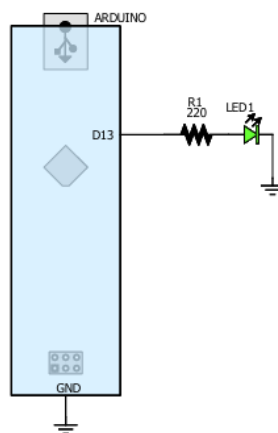
6. ลำดับขั้นการปฏิบัติงาน

1. ศึกษาจุดประสงค์ทั่วไปและทฤษฎีพื้นฐานที่เกี่ยวข้อง
2. ดำเนินการต่อวงจรบอร์ดทดลองตามวงจรที่กำหนด
3. เขียนโปรแกรมควบคุมและทดสอบการทำงานของวงจร
4. สรุปผลการปฏิบัติงาน

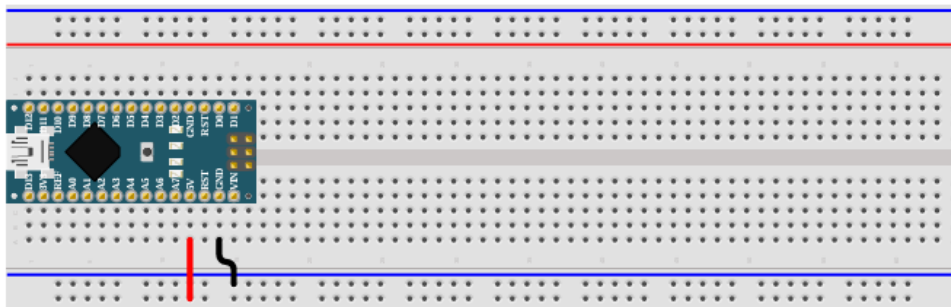
7. วงจรที่ใช้ทดลอง

1. วงจรไมโครคอนโทรลเลอร์ Arduino ที่ใช้บอร์ดโมดูล Arduino สำเร็จรูป
2. ทดลองด้วยการจำลองการทำงานด้วยโปรแกรม Proteus

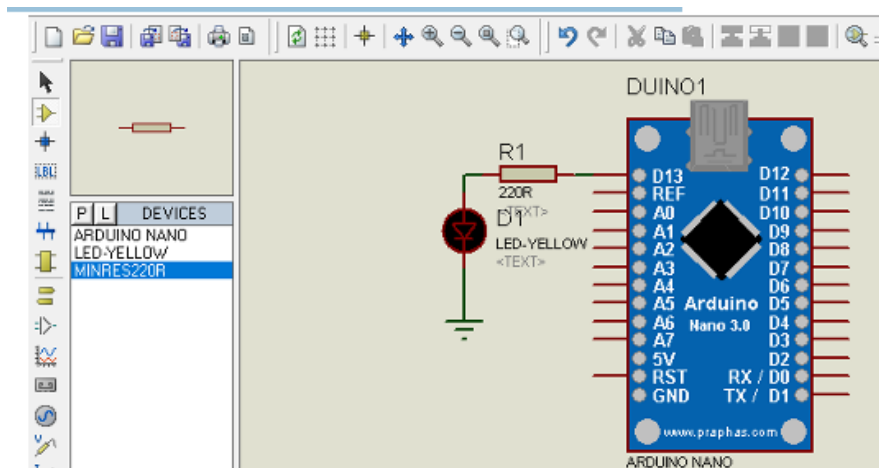
กรณีที่ใช้ Arduino ในการทดลอง ต่อดังรูป



การต่อวงจรไมโครคอนโทรลเลอร์ Arduino ที่ใช้บอร์ดโมดูล Arduino สำเร็จรูป



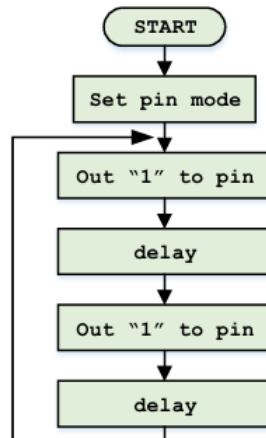
การต่อวงจรเพื่อทดลองด้วยการจำลองการทำงานด้วยโปรแกรม Proteus



8. การเขียนโค้ดโปรแกรมควบคุม

1. กำหนดชื่อตัว LED กับขาพอร์ตที่ต้องใช้งาน	5. ส่งค่า LOW ไปยังขาพอร์ตเพื่อให้ LED ดับ
2. กำหนดโหมดขาที่เชื่อมต่อ LED	6. หน่วงเวลา
3. ส่งค่า HIGH ไปยังขาพอร์ตเพื่อให้ LED ติด	7. วนกลับไปทำลำดับที่ 3 ซ้ำ
4. หน่วงเวลา	

แปลงลำดับงานเป็นผังงาน จากลำดับงานสามารถเขียนเป็นผังงานได้ดังนี้



แปลงผังงานเป็นโปรแกรม จากผังงานสามารถเขียนเป็นโค้ดโปรแกรมควบคุม Arduino

```

1  #define LED 13
2  void setup()
3  {
4      pinMode(LED, OUTPUT);
5  }
6  void loop()
7  {
8      digitalWrite(LED, HIGH); // turn the LED on (HIGH is the voltage level)
9      delay(1000); // wait for a second
10     digitalWrite(LED, LOW); // turn the LED off by making the voltage LOW
11     delay(1000); // wait for a second
12 }
  
```

รายละเอียดโค้ดโปรแกรม

- บรรทัดที่ 1 ประกาศชื่อ LED แทนตัวเลข 13
- บรรทัดที่ 4 กำหนดโหมดของขาพอร์ตที่เชื่อมต่อ LED ให้ทำงานเป็นเอาต์พุตพอร์ต
- บรรทัดที่ 8 ส่งค่าลอจิก 1 ออกจากขาพอร์ตที่ต่อ LED
- บรรทัดที่ 9 หน่วงเวลา 1 วินาที (1000 ms)
- บรรทัดที่ 10 ส่งค่าลอจิก 0 ออกจากขาพอร์ตที่ต่อ LED
- บรรทัดที่ 11 หน่วงเวลา 1 วินาที (1000 ms)

9. สรุปและวิจารณ์ผล

นักเรียนต้องได้ผลการประเมินไม่ต่ำกว่าร้อยละ 60 และหากไม่ผ่านเกณฑ์การประเมินจะต้อง ปฏิบัติงานใหม่

.....

.....

10. การประเมินผล

ใบประเมินผลวิชา ไมโครคอนโทรลเลอร์		ค่าน้ำหนัก	ค่าน้ำหนัก
หน่วยที่ ชื่อหน่วย.....		คะแนน	คะแนนที่
เรื่อง.....		หมวดวัด	แท้จริง
ชื่อ-สกุล.....ชั้น.....เลขที่.....		(คะแนน)	(คะแนน)
ลำดับที่	รายการแบบประเมิน		
1.	แบบฝึกหัด	30	3
2.	ใบบาง - การเตรียมเก็บรักษาเครื่องมือ (10) - ทักษะในการปฏิบัติงาน (20) - ปฏิบัติงานถูกต้องตามขั้นตอน (15) - ความถูกต้องของใบบาง (15)	60	6
3.	คุณธรรมจริยธรรม - ความตรงต่อเวลา (2) - การแต่งกาย (2) - ความตั้งใจในการปฏิบัติงาน (2) - การทำงานร่วมกับผู้อื่น (2) - ความรับผิดชอบ (2)	10	1
คะแนนรวมที่ได้		100	10

ข้อเสนอแนะ.....

ลงชื่อ.....ผู้ประเมิน

(.....)

...../...../.....

แบบบันทึกสรุปผลคะแนนประจำวิชาไมโครคอนโทรลเลอร์

หน่วยที่.....ชื่อหน่วย.....

รายการแบบประเมิน		แบบฝึกหัด	ใบงาน	คุณธรรมจริยธรรม	รวมคะแนนที่ได้
ค่าน้ำหนักคะแนนที่แท้จริง		3	6	1	10
เลขที่	ชื่อ-สกุล	คะแนนที่ได้			
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					

11. เอกสารอ้างอิง /เอกสารค้นคว้าเพิ่มเติม

บรรณานุกรม ภาษาไทย ไมโครคอนโทรลเลอร์ : ศูนย์ส่งเสริมอาชีพฯ, 2567.

(สาโรช กล้ามอญ, 2562)