



SignGlove

(Hand sign language translator glove)

จัดทำโดย

นาย คุณานนต์ กัมมั่น เลขที่ 1 ม.6/10

นาย พรีพัฒน์ ประเสริฐศิริ เลขที่ 3 ม.6/10

นาย กฤชดินทร์ โสสีغا เลขที่ 10 ม.6/10

นาย อุ้มบุญ หนี้เชสงค์ เลขที่ 13 ม.6/10

นาย จารย์วรวรรณ หัสดุณ เลขที่ 15 ม.6/10

นาย ณัฐธนนท์ วิถิตร้านนท์ เลขที่ 16 ม.6/10

นาย ธนาท พชร์พันธุ์ ธนาคม เลขที่ 17 ม.6/10

นาย นรบดี ถิรพัฒน์กุล เลขที่ 18 ม.6/10

อาจารย์ที่ปรึกษา

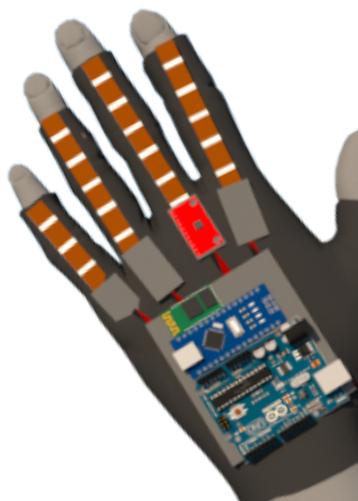
ครูญาดา เดิมสมบูรณ์

รายวิชา โครงการ IS (ว30201)

ปีการศึกษา 2566

โรงเรียนเบญจมราษฎร์ทิศ จังหวัดจันทบุรี

Benchamarachuthit Chanthaburi School



บทคัดย่อ

โครงการนี้มีจุดประสงค์ในการสร้าง อุปกรณ์เพื่อช่วยเหลือผู้พิการทาง การได้ยิน, การพูด ด้วย arduino และ ภาษา C ในทางด้านการสื่อสารด้วยภาษาเมืองไทยให้มีความสะดวกต่อการสนทนาอย่างขึ้น นำเสนອออกแบบมาในรูป ถุงมืออัจฉริยะ ที่สามารถใช้สื่อสาร กับผู้คนใหม่ในปกติได้ ใช้งานผ่านการสัมมัสโดยใช้ผู้ใช้ทำภาษาเมือง และอุปกรณ์จะแปลงความหมายออกมายเป็นเสียง และมีการรับเสียงจากคู่สนทนา มาเป็นข้อความบนจอเมือถือ หรือ จอ led บนอุปกรณ์ ตลอดจนการประยุกต์ใช้ พัฒนาระบบการทำงานโดย ภาษา C ออกแบบโดย blender หรือ sketchup pro และทดสอบการทำงานของระบบด้วย tinkercad ตัวช่วยในการทดลองการใช้จริงของ arduino board ถึงแม้จะไม่ได้ยิน พูดไม่ได้ แต่ก็ไม่มีข้อกำหนดใดในโลก ที่ไม่ให้เราสื่อสารกัน

กิตติกรรมประกาศ

โครงการ SignGlove (Hand sign language translator glove) สำเร็จลุล่วงได้ดีนี้เนื่องจากการสนับสนุนจาก ผู้มีค

วางแผนที่ดีที่สุด ที่ได้ผลลัพธ์ที่ดีที่สุด ให้กับผู้ใช้งาน ไม่ว่าจะเป็นเด็ก หรือผู้สูงอายุ ที่ต้องการเข้าใจสิ่งที่อยู่รอบตัว ไม่ว่าจะเป็นภาษาไทย ภาษาอังกฤษ หรือภาษาอื่นๆ ที่ต้องการเข้าใจ ไม่ว่าจะเป็นผู้ใช้งานที่มีความสามารถทางด้านเทคโนโลยีสูง หรือผู้ใช้งานที่มีความสามารถทางด้านเทคโนโลยีต่ำ แต่ทั้งหมดนี้ สามารถทำได้โดยใช้เทคโนโลยีที่มีอยู่แล้ว ไม่ต้องซื้ออุปกรณ์ที่ราคาแพง หรือซับซ้อนมาก แต่เพียงแค่ติดตั้งซอฟต์แวร์ที่เหมาะสม ก็สามารถใช้งานได้แล้ว

- ขอขอบคุณ ครูภูมิ ที่ช่วยให้คำปรึกษาและแนะนำแนวทาง ในการแก้ไขข้อบกพร่องต่างๆ ให้สำเร็จ ลุล่วงไปได้ด้วยดี
- ขอขอบคุณ เพื่อนๆ ที่ร่วมช่วยงานทุกคนที่อำนวยความสะดวกและช่วยเหลือในการทำงานครั้งนี้

คำนำ

โครงการฉบับนี้ เป็นส่วนหนึ่งของวิชา I30201 โครงงาน IS ชั้นมัธยมศึกษาปีที่ ๕ โดยมีจุดประสงค์ เพื่อศึกษา ช่วยเหลือผู้พิการทางการได้ยิน การพูด และ เพื่อการศึกษาความรู้ในเรื่อง arduino และ ภาษา C ซึ่งรายงานฉบับนี้มีเนื้อหา ขั้นตอนการทำอุปกรณ์ (Hand sign language translator glove) กับการศึกษาปัญหาทางการได้ยิน การพูด ของผู้พิการ

และ ภาษาไม้อ ทำให้ผู้พิการที่สามารถใช้สื่อสาร กับผู้คนหรืออนปกติได้ ใช้งานผ่านการสัมภาร์ และเมื่อทำภาษาไม้อจะเปล่งความหมายออกมาเป็นเสียง และการรับเสียงจากผู้สนทนามาเป็นข้อความบนจอ มีถือ หรือ จอ led บนอุปกรณ์ ตลอดจนการประยุกต์ใช้

ผู้จัดทำโครงงานได้เลือกหัวข้อนี้ในการทำโครงงาน เนื่องจากเป็นเรื่องที่น่าสนใจ และ สามารถทำออกมาเพื่อช่วยเหลือผู้พิการได้จริง ซึ่งมีการรับรองและทดสอบให้เป็นตัวอย่างให้ดูมาแล้ว ผู้จัดทำจะต้องขอขอบคุณ อาจารย์ที่ปรึกษา ผู้ให้ความรู้ และแนวทางการศึกษาเพื่อน ๆ ทุกคนที่ให้ ความช่วยเหลือมาโดยตลอด ผู้จัดทำหวังว่ารายงานฉบับนี้จะให้ความรู้ และเป็นประโยชน์แก่ผู้อ่านทุก ๆ ท่าน ขอบคุณครับ

สารบัญ

เรื่อง

หน้า

คำนำ

3

สารบัญ

4

บทที่ 1 บทนำ

7

หลักการและเหตุผล	7
จุดประสงค์	7
ขอบเขตและการทำงาน	7
แผนการดำเนินงาน	8
ประโยชน์ที่ได้รับ	8
อุปกรณ์และเครื่องมือที่ใช้	8
บทที่ 2 ความรู้พื้นฐานและทฤษฎีที่เกี่ยวข้อง	9
บุคคลที่มีความบกพร่องทางการได้ยิน	9
บุคคลที่มีความบกพร่องทางการพูดและภาษา	9
ภาษา C	9
อุปกรณ์ Arduino	43
สารบัญ	
เรื่อง	หน้า
บทที่ 3 วิธีดำเนินงาน	52
ขั้นตอนการดำเนินงาน	52
วิธีการรวมข้อมูล	52

อุปกรณ์และเครื่องมือที่ใช้	52
แผนผังอธิบายการทำงาน	53
การดำเนินงาน	56
บทที่ 4 วิธีดำเนินงาน	59
ผลการพัฒนาโครงงาน	59
ตัวอย่างผลงาน	59
ผลการประเมินโครงงาน	60
บทที่ 5 สรุปผลการดำเนินงาน	62
การดำเนินงานจัดทำโครงงาน	62
ผลที่ได้รับ ปัญญา และ อุปสรรค	62
ข้อเสนอแนะต่อโครงงาน	63
สารบัญ	
เรื่อง	หน้า
บทที่ 5 สรุปผลการดำเนินงาน	62
สรุปผลการดำเนินงานโครงงาน	63
ภาคผนวก	64

ก.แบบจำลองด้วยเร็บ www.tinkercad.com	64
ข.แบบจำลองด้วยเร็บ OpenCV	66
ค.โมเดลสามมิติ	67

บทที่ 1 บทนำ

โครงการ “ SignGlove (Hand sign language translator glove) ”

1.1 หลักการและเหตุผล

ในยุคสมัยนี้ที่การสื่อสารเป็นสิ่งสำคัญ การพัฒนา SignGlove คือการไม่แบ่งแยกการสื่อสาร ด้วยการสร้างอุปกรณ์ที่ช่วยให้ผู้พิการทางการได้ยิน และ การพูดสามารถโต้ตอบกับผู้อื่นได้อย่างราบรื่น เราส่งเสริมความเสมอภาคและการเข้าถึง การทำลายอุปสรรคในการสื่อสารช่วยส่งเสริมสังคมที่มีความครอบคลุมมากขึ้น SignGlove ช่วยให้ผู้พิการมีวิธีในการแสดงออกโดยตรงและเป็นอิสระ ช่วยให้พูดเข้าไม่ต้องพึงพาลามด้วยการส่งเสริมการไม่แบ่งแยกการสื่อสาร ช่วยให้ผู้พิการมีความมั่นใจและกล้าแสดงออกในสังคมและอาชีพต่างๆ

Arduino เป็นแพลตฟอร์มอิเล็กทรอนิกส์ ทำหน้าที่เป็นสมองของ SignGlove โดยจะประมวลผลข้อมูลจากเซ็นเซอร์และใช้อัลกอริทึมที่ซับซ้อนเพื่อดำเนินการเคลื่อนไหวของมือเฉพาะที่เกี่ยวข้องกับตัวอักษร คำ และสัญลักษณ์ภาษาไทยแต่ละรายการ มีจุดมุ่งหมายเพื่อให้ใช้

งานง่ายและใช้งานง่าย สามารถเชื่อมต่อกับคอมพิวเตอร์หรืออุปกรณ์พกพาที่ประมวลผลข้อมูล ระบบ呢 และยังได้รับการออกแบบมาให้เข้าถึงได้ง่าย ทำให้สามารถสื่อสารได้อย่างราบรื่นในสถานการณ์ต่างๆ

SignGlove ได้ใช้หลักการของการไม่แบ่งแยกทางการสื่อสาร และการนำ Arduino มาประยุกต์ใช้ โดยมีเป้าหมายเพื่อทลายอุปสรรคในการสื่อสารสำหรับผู้พิการ ด้วยการให้การเข้าถึงการสื่อสารที่ดีขึ้นและส่งเสริมความเป็นอิสระ ด้วยเทคโนโลยีที่ก้าวล้ำในปัจจุบันมีศักยภาพในการเปลี่ยนแปลงชีวิตของผู้คน ทำให้มั่นใจได้ว่า เสียงของทุกคนจะได้รับการรับฟังและมีคุณค่า ด้วย การนำ SignGlove และเทคโนโลยีมาช่วยเหลืออย่างแพร่หลาย

1.2 จุดประสงค์

1.2.1 เพื่อศึกษาเกี่ยวกับความบกพร่องทางการได้ยิน, การพูด และค้นคว้าหาทางช่วยให้สามารถสื่อสารอย่างปกติได้สะดวก

1.2.2 เพื่อศึกษาการใช้ภาษาเมือง

1.2.3 เพื่อพัฒนาผู้เรียนให้มีความสามารถทาง การคิด, ตระกะ เพิ่มขึ้นเนื่องจากการเขียนการทำงานของระบบ ด้วยภาษาC/C++ และบอร์ด arduino เป็นสิ่งที่เคยศึกษามาก่อนจึงอยากรู้เพิ่มขึ้นซึ่งเป็นการพัฒนาตนเอง

1.3 ขอบเขตการทำงาน

1.3.1 ใช้ในการสื่อสารด้วยภาษามือ

1.3.2 สามารถแปลงความหมายอ้อมปากเป็นเสียง และรับเสียงมาเป็นข้อความได้

1.3.3 ผู้ใช้จำเป็นต้องสวมใส่อุปกรณ์ และมีแอปสำหรับแสดงผล

1.3.4 รองรับภาษาไทย, ภาษาอังกฤษ เท่านั้น

1.4 แผนการดำเนินงาน

หัวข้อ/เดือน	มิ.ย. 66	ก.ค. 66	ส.ค. 66	ก.ย. 66	ต.ค. 66	พ.ย. 66
การศึกษาความรู้ที่เกี่ยวกับโครงงาน						
ออกแบบอุปกรณ์ การทำงาน และแผนโครงงาน						
ทดสอบระบบการทำงาน และเตรียมอุปกรณ์						
ทำอุปกรณ์ และทดลองใช้จริง						
ตรวจสอบและแก้ไขปัญหาของระบบการทำงาน						

1.5 ประโยชน์ที่ได้รับ

1.5.1 ช่วยเหลือ ผู้พิการทางการได้ยิน, การพูด ให้สามารถสื่อสารอย่างปกติกับผู้คน ได้สะดวกยิ่งขึ้น และรู้ท่าทางภาษามือ

1.5.2 พัฒนาตนเองให้มีความสามารถมากขึ้น เนื่องจากการศึกษา arduino และ ภาษาC จะช่วยทำให้เกิดความรู้ใหม่ๆ สำหรับตนเอง

1.6 อุปกรณ์และเครื่องมือที่ใช้

1.6.1 โปรแกรมที่ใช้ Arduino IDE, visual studio, visual code, Blender, Sketchup ในการทำงาน แต่ละโปรแกรมมีหน้าที่ต่างนี้

- Arduino IDE ใช้ในการเขียนโค้ดระบบการทำงาน
- visual studio, visual code, ในการสร้างแอปและผลการทำงาน
- Blender, Sketchup ในการสร้างโมเดลสามมิติ

1.6.2 เครื่องมือที่ใช้แสดงผล โทรศัพท์มือถือ และ จอLED

1.6.3 อุปกรณ์ใช้ทำ โทรศัพท์มือถือ และ คอมพิวเตอร์ที่ดังนี้

- บอร์ด Arduino (ชิ้น Arduino Uno หรือ Arduino Nano)
- Flex sensors (เพื่อวัดค่าการโค้งของนิ้ว)
- gyroscope sensor หรือ accelerometer sensors (เพื่อตรวจจับการเคลื่อนไหวของมือ)
- โมดูล Bluetooth (สำหรับการสื่อสารไร้สาย เช่น HC-05 หรือ HC-06)
- โมดูล ไมโครโฟน (สำหรับการรับเสียง)
- ลำโพง หรือ ออด (สำหรับปล่อยเสียง)
- จอแสดงผล LCD (สำหรับแสดงผลตัวอักษรหรือข้อความ)
- ถุงมือ (ควรถอดปลายนิ้วออกเพื่อติดเชือกช่อง)
- โทรศัพท์บอร์ด และสายจัมเปอร์ - ด้วย หรือ ลวดนำไฟฟ้า
- แบตเตอรี่หรือแหล่งพลังงาน - ตัวต้านทาน (หากใช้เซนเซอร์แบบเฟลิกซ์)

บทที่ 2 ความรู้พื้นฐานและทฤษฎีเกี่ยวกับ

2.1 บุคคลที่มีความบกพร่องทางการได้ยิน

บุคคลที่มีความบกพร่องทางการได้ยิน ได้แก่ บุคคลที่สูญเสียการได้ยินตั้งแต่ระดับทึบตึงน้อยจนถึงทุหันวาก ซึ่งแบ่งเป็น 2 ประเภท ดังนี้

2.2.1 คนทุหันวาก หมายถึง บุคคลที่สูญเสียการได้ยินมากจนไม่สามารถเข้าใจการพูดผ่านทางการได้ยินไม่ว่าจะใส่หูอีก多么เสียเครื่องช่วยฟัง ซึ่งโดยทั่วไปหากต้องการได้ยินจะมีการสูญเสียการได้ยิน 90 เดซิเบลขึ้นไป

2.2.2 คนทึบตึง หมายถึง บุคคลที่มีการได้ยินเหลืออยู่เพียงพอที่จะได้ยินการพูดผ่านทางการได้ยิน โดยทั่วไปจะใส่เครื่องช่วยฟัง ซึ่งหากตรวจวัดการได้ยินจะมีการสูญเสียการได้ยินน้อยกว่า 90 เดซิเบลลงมาถึง 26 เดซิเบล

2.2 บุคคลที่มีความบกพร่องทางการพูดและภาษา

บุคคลที่มีความบกพร่องทางการพูดและภาษา ได้แก่ บุคคลที่มีความบกพร่องในการเปล่งเสียงพูด เช่น เสียงผิดปกติ อัตราความเร็วและจังหวะการพูดผิดปกติ หรือบุคคลที่มีความบกพร่อง ในเรื่องความเข้าใจหรือการใช้ภาษาพูด การเขียนหรือระบบสัญลักษณ์อื่นที่ใช้ในการติดต่อสื่อสาร ซึ่งอาจเกี่ยวกับรูปแบบ เนื้อหาและหน้าที่ของภาษา

2.3 ภาษา C/C++

ภาษา C เป็นภาษาคอมพิวเตอร์ที่อวัตถุประสงค์ทั่วไป เป็นภาษาที่มีความจำเป็นมาก มั่นคงสนับสนุนการเขียนโปรแกรมที่มีโครงสร้าง การกำหนดขอบเขตของตัวแปร และการเรียกใช้ตัวเอง (Recursion) และมั่นเป็นภาษาที่อยู่ในระดับต่ำ (Low level) นั่นคือ มันเป็นภาษาที่สามารถทำงานได้ดีในระดับของชาร์ดแวร์ ภาษา C เป็นสามารถที่ออกแบบมาให้สามารถที่จะทำงานกับคำสั่งพื้นฐานของคอมพิวเตอร์ได้อย่างมี

ประสิทธิภาพ เพราะฉะนั้นมันจึงถูกพบบ่อยในการใช้สร้างแอพพลิเคชันในสมัยก่อนที่เขียนโดยภาษาและเซนเบลี รวมถึงระบบปฏิบัติการ เช่น เดียวกันกับซอฟต์แวร์ประยุกต์สำหรับคอมพิวเตอร์ ซุปเปอร์คอมพิวเตอร์ และระบบฝึกตัว

ภาษา C นั้นถูกพัฒนาครั้งแรกโดย Dennis Ritchie ในระหว่างปี 1969 และ 1973 ที่ Bell Labs และใช้สำหรับพัฒนาและปรับปรุงระบบปฏิบัติการ Unix ใหม่ ตั้งแต่นั้นมันได้มาเป็นภาษาที่มีการใช้งานอย่างกว้างขวางมากที่สุดตลอดเวลา ที่มา กับ C คอมไฟเลอร์จากบริษัทพัฒนาต่างๆ สำหรับพัฒนานิสตานิตยกรรมคอมพิวเตอร์และระบบปฏิบัติการที่ได้รับความนิยมเป็นจำนวนมาก ภาษา C ได้ถูกกำหนดมาตรฐานโดย American National Standards Institute (ANSI) ตั้งแต่ปี 1989 และ International Organization for Standardization (ISO) ในเวลาต่อมา

2.3.1 โครงสร้างของโปรแกรม

2.3.1.1 คำศัพท์ของภาษา C โดยปกติแล้วโปรแกรมของผู้เริ่มต้นเป็นโปรแกรมที่เรียกว่า "Hello world" ซึ่งจะแสดงคำว่า "Hello World!" ออกทางจอคอมพิวเตอร์ของคุณ มันเป็นโปรแกรมที่ง่ายที่สุดที่จะแสดงให้นักพัฒนามือใหม่ได้เห็น มาดูว่ามันเป็นอย่างไร

```
#include <stdio.h>
int main()
{
    printf("Hello world!\n");
    return 0;
}
```

โปรแกรมนี้เป็นโปรแกรมเพื่อแสดงข้อความอย่างง่าย "Hello world!" ออกทางหน้าจอ เราได้ใช้ฟังก์ชันต่างๆ จากไลบรารี stdio.h ของภาษา C ที่สามารถให้เราทำสิ่งต่างๆ ได้ และแสดงข้อความออกทางจอภาพหรือรับค่าจากคีย์บอร์ด

Hello world!

นี่เป็นผลลัพธ์การทำงานของโปรแกรมซึ่งจะแสดงผลข้อความ "Hello world!" ออกทางหน้าจอ คุณสามารถเปลี่ยนเป็นชื่อของคุณเองแล้วดูผลลัพธ์

2.3.1.2 Blocks (บล็อก) คือสิ่งที่กำหนดขอบเขตและควบคุมการทำงานของโปรแกรม ซึ่งจะใช้เครื่องหมาย { และสิ้นสุดด้วย } ในภาษา C บล็อกนั้นมีหลายรูปแบบ เช่น บล็อกของฟังก์ชัน บล็อกของคำสั่งควบคุม หรือบล็อกอยู่ในโปรแกรม และนอกจากนี้บล็อกยังสามารถซ้อนกันได้ นี่เป็นตัวอย่างของบล็อกในภาษา C

```
#include <stdio.h>
int sum(int a, int b)
{
    return a + b;
}
int main()
{
    int x = 3, y = 4;
```

```

if (x < y)

{
    printf("%d\n", sum(x, y));
}

printf("Summation program\n", sum(x, y));

return 0;
}

```

ในตัวอย่าง เป็นโปรแกรมหาผลรวมของตัวเลขสองจำนวนในภาษา C ในโปรแกรมนี้ประกอบไปด้วย 3 บล็อก คือบล็อกของฟังก์ชัน main ซึ่งเป็นฟังก์ชันหลักในการทำงานของโปรแกรมซึ่งโปรแกรมจะเริ่มการทำงานที่นี่ บล็อกที่สองเป็นของฟังก์ชัน sum() ซึ่งเป็นฟังก์ชันสำหรับหาผลรวมของตัวเลขที่เรียกโดยฟังก์ชันหลัก และภายในฟังก์ชัน main มีบล็อกคำสั่งเงื่อนไข if เพื่อควบคุมการทำงานของโปรแกรมอยู่ภายใน

```
#include <stdio.h>
```

ในภาษา C คำสั่ง #include ใช้สำหรับนำเข้าโค้ดจากไฟล์อื่นเข้ามาใช้งานในโปรแกรม โดยปกติแล้วเราได้นำเข้ามาโดยไฟล์ stdio.h ของภาษา C ซึ่งประกอบไปด้วยฟังก์ชันต่างๆ ที่จำเป็นในการเขียนโปรแกรม เช่น ฟังก์ชันสำหรับแสดงผลออกทางหน้าจอ หรือรับค่าจากคีย์บอร์ด เป็นต้น คุณสามารถสร้างไฟล์ .h ของคุณเองได้เพื่อนำมาใช้ในโปรแกรม

2.3.1.3 Comment เป็นส่วนของโค้ดที่ไม่มีผลต่อการการทำงานของโปรแกรม มันถูกใช้เพื่ออธิบายโปรแกรมสำหรับมนุษย์เข้าใจ ในภาษา C เราสามารถคอมเม้นท์ได้สองวิธี คือ การคอมเมนต์แบบหนึ่งบรรทัดและแบบหลายบรรทัด

```
// Single line comment example

#include <stdio.h>

int main()

{
    // This is my comment

    printf("Hello C language\n");

    return 0;
}
```

ในตัวอย่างเป็นการคอมเมนต์แบบหนึ่งบรรทัด เราจะใช้เครื่องหมาย Double slash (//) และตามด้วยสิ่งที่เราต้องการคอมเมนต์ ในตัวอย่างเราได้สร้างคอมเม้นท์สองอันในโปรแกรม

```
/* Multiple lines comment example */

#include <stdio.h>

int main()

{
    /* This is my comment

    This another line comment */

    printf("Hello C language\n");

    return 0;
}
```

```
}
```

คอมเม้นท์อีกแบบหนึ่งคือการคอมเมนต์หลายบรรทัด มันจะลงทะเบียนทุกอย่างหลังจากการประกาศครั้งแรกของเครื่องหมาย Slash star (*) และสิ้นสุดที่ Star slash (/)

2.3.1.4 เซมิโคลอน (Semicolon) (;) เป็นสัญลักษณ์ที่ใช้เพื่อแบ่งแยกคำสั่งภายในโปรแกรม ซึ่งมันหมายถึงการจบคำสั่นนั้นๆ เซมิโคลอนนี้ใช้ในภาษาต่างๆ และมันเป็นสิ่งที่บังคับ เพื่อให้ตัวคอมไพล์อร์ของภาษาสามารถแยกแยะคำสั่งในการทำงานได้

```
int a;
int b = 5; b = 2;
printf("%d", a + b);
```

ในตัวอย่างเรามีคำสั่ง บรรทัดแรกเป็นการประกาศตัวแปร บรรทัดที่สองคุณจะเห็นสองคำสั่งอยู่ในบรรทัดเดียวกันและมันสิ้นสุดด้วยเซมิโคลอน บรรทัดที่สามจะแสดงค่าผลรวมของตัวแปร a และ b

2.3.1.5 Whitespace คือตัวอักษรหรือเครื่องหมายที่ใช้แบ่งแยกคำสั่งและ Token ในเค็ดของโปรแกรม ในภาษา C นั้น Whitespace จะประกอบไปด้วย การเว้นวรรค Tab และการขึ้นบรรทัดใหม่ Whitespace ที่เรียกว่าเป็นจำนวนมากนั้นไม่มีผลต่อการทำงานของโปรแกรม และคอมไพล์อร์ แต่มันช่วยให้โปรแกรมเมอร์สามารถทำได้โดยที่ไม่ต้องใส่จมูกและสามารถอ่านเข้าใจได้ง่ายขึ้นโดยคนอื่นๆ แต่ Whitespace ยังคงต้องใช้ในบางที่ เช่น ระหว่างคำสั่งของภาษา C และชื่อของตัวแปร เป็นต้น

```
int a = 1;
int b = 2;
int c = 3;
```

ในตัวอย่าง เป็นการใช้ Whitespace ในรูปแบบต่างๆ ซึ่งถือว่าเป็นรูปแบบการเขียนโปรแกรมของแต่ละคน แต่โดยปกติแล้วเรา尼ยมใช้ Whitespace ในรูปแบบของการประกาศตัวแปร a

2.3.1.6 Keyword เป็นกลุ่มคำที่ถูกสงวนไว้โดยเราไม่สามารถใช้คำเหล่านี้ในการประกาศเป็นชื่อตัวแปร ฟังก์ชัน ซึ่งในโปรแกรมทุกภาษาต่างก็มี Keyword

นี่เป็น Keyword มาตรฐานในภาษา C

auto	break	case	char
const	continue	default	do
double	else	enum	extern
float	for	goto	if
int	long	register	return

short	signed	sizeof	static
struct	switch	typedef	union
unsigned	void	volatile	while

Keyword เหล่านี้มีหน้าที่การทำงานที่แน่นอนซึ่งขึ้นกับวัตุประสงค์ของมัน int, short, float, double ใช้เพื่อประกาศตัวแปร ในขณะที่ if for while เป็นคำสั่งในการควบคุมการทำงานของโปรแกรม

2.3.2 ตัวแปรและประเภทข้อมูล

2.3.2.1 ตัวแปรในภาษา C ตัวแปรเป็นสถานที่สำหรับเก็บข้อมูลในหน่วยความจำ โดยมีชื่อของตัวแปรเพื่อใช้อ้างถึงข้อมูล (Identifier) ตัวแปรถูกใช้เพื่อกำกับค่า เราสามารถสร้างตัวแปรได้เป็นจำนวนมากโดยมีชื่อที่แตกต่างกัน รูปแบบของการประกาศตัวแปรในภาษา C คือ:

```
type identifier;
```

```
type identifier = value;
```

type เป็นประเภทของข้อมูลที่จะเก็บในตัวแปร identifier เป็นที่รู้จักกันในในชื่อของตัวแปร เราใช้ชื่อนี้เพื่ออ้างถึงค่าที่ตัวแปรนั้นเก็บอยู่ values เป็นทางเลือกที่คุณสามารถกำหนดค่าให้กับตัวแปรเมื่อมันถูกสร้าง หรือกำหนดในภายหลังได้ มาดูตัวอย่าง

```
int a;
```

```
float b;
```

```
char c = 'A';
```

ในตัวอย่างเรามีตัวแปร 3 ตัว ตัวแรกประเภทของมันคือ int และมีชื่อว่า a มันใช้เพื่อกำกับค่าเลขจำนวนเต็ม (Integer) และเรายังไม่ได้กำหนดค่าใดๆ ให้มัน ดังนั้นค่าเริ่มต้นของตัวแปรเมื่อถูกสร้างขึ้นจะเป็น NULL ตัวแปรที่สองมีประเภทเป็น float ตัวแปรนี้จะถูกใช้เพื่อกำกับค่าของจำนวนจริง และตัวแปรที่สามมีประเภทเป็น char มันถูกใช้เพื่อกำกับค่าลักษณ์หนึ่งตัวใน ASCII code และเรากำหนด 'A' เป็นค่าเริ่มต้นให้กับมัน

2.3.2.2 ประเภทข้อมูล ในภาษา C มีประเภทข้อมูลชนิดต่างๆ ที่ให้สามารถใช้เพื่อชนิดการกับข้อมูลต่างประเภทกัน เช่น Boolean, Number, Character และ Object เป็นต้น

ตารางข้างล่างนี้แสดงประเภทข้อมูลทั้งหมดในภาษา C

Data type	Size	Value
char	1 byte	-128 to 127
unsigned char	1 byte	0 to 255

signed char	1 byte	-128 to 127
int	4 byte	-2,147,483,648 to 2,147,483,647
unsigned int	4 byte	0 to 4,294,967,295
short	2 byte	-32,768 to 32,767
unsigned short	2 byte	0 to 65,535
long	4 byte	-2,147,483,648 to 2,147,483,647
unsigned long	4 byte	0 to 4,294,967,295
float	4 byte	1.2E-38 to 3.4E+38
double	8 byte	2.3E-308 to 1.7E+308
long double	10 byte	3.4E-4932 to 1.1E+4932
bool	1 bit	0 to 1

2.3.2.3 การประกาศตัวแปร การประกาศและกำหนดค่าเริ่มต้นให้กับตัวแปร เราจะยกตัวอย่างง่ายๆ ดังนี้

```
#include <stdio.h>

int main()
{
    int a = 5;
    int b;
    b = 10;
    int c = a + b;
    printf("Value of c is %d", c);
    return 0;
}
```

ในตัวอย่าง เราได้สร้างตัวแปรสามตัวคือ a, b และ c ที่มีชนิดข้อมูลเป็น int และเรารวมค่าของ a และ b แล้วเก็บไว้ในตัวแปร c ข้างล่างนี้เป็นผลลัพธ์ของโปรแกรม

Value of c is 15

อย่างไรก็ตาม คุณสามารถประกาศชนิดข้อมูลอื่นได้ ซึ่งมันไม่อยากเกินไป มาดูตัวอย่างเพิ่มเติม

```
#include <stdio.h>
int main()
{
    float PI = 3.14;
    int r;
    scanf("Enter circle radius %d", &r);
    double result = 2 * PI * r;
    printf("Area is %f", result );
    return 0;
}
```

ในตัวอย่าง แสดงให้คุณเห็นการใช้ตัวแปรประเภทต่างๆ ในภาษา C นั้นมีข้อมูลประเภทต่างๆ ที่ให้ความสามารถจัดการกับข้อมูลที่แตกต่างกันออกไป เช่น ตัวเลขจำนวนเต็ม ตัวเลขจำนวนจริง หรือข้อมูลแบบตัวอักษร ซึ่งข้อมูลแต่ละประเภทมีรูปแบบการใช้งานและขนาดของข้อมูลที่แตกต่างกัน ดังแสดงในตารางก่อนหน้า ต่อไปเราจะอธิบายและแสดงตัวอย่างของการใช้ข้อมูลประเภทต่างๆ ในภาษา C สำหรับเขียนโปรแกรม

2.3.2.3 Boolean เป็นประเภทข้อมูลที่มีได้เพียงสองค่าคือจริง true และเท็จ false ซึ่งเป็นประเภทของตัวแปรสำหรับเก็บค่าที่เป็นไปได้เพียงสองค่า เช่น กลางวันและกลางคืน สีขาวและสีดำ ร้อนและหนาว เป็นต้น ข้อมูลประเภทนี้จะใช้สำหรับสร้าง Expression สำหรับเงื่อนไข การทำงานของโปรแกรม ในภาษา C ค่าของ Boolean นั้นใช้ 1 หรือ true สำหรับค่าที่เป็นจริงและ 0 หรือ false สำหรับค่าที่เป็นเท็จ มาดูตัวอย่างการใช้งาน

```
#include <stdio.h>
#include <stdbool.h>
int main ()
{
    bool gender = 1;
    bool open = 0;
    if (gender)
    {
        printf("This is the boy.\n");
    }
    else
    {
```

```

    printf("This is the girl.\n");

}

if (open)

{

    printf("The door is open.\n");

}

else

{

    printf("The door is not open.\n");

}

return 0;

}

```

ในตัวอย่าง เป็นการใช้งานตัวแปรประเภท Boolean ในภาษา C ในตัวแปร gender เราได้กำหนดตัวแปรให้หมายถึงเพศ ถ้าในตัวแปรนั้นมีค่าเป็นจริงจะหมายถึงเป็นผู้ชาย และถ้ามีค่าเป็นเท็จจะหมายถึงเป็นผู้หญิง และตัวแปร open ใช้สำหรับเก็บสถานะการเปิดและปิดของประตู เมื่อตัวแปรมีค่าเป็นจริงจะหมายถึงประตูถูกเปิดอยู่ และเมื่อมีค่าในตัวแปรเป็นเท็จก็หมายความว่าประตูนั้นไม่ได้เปิด

This is the boy.

The door is not open.

ในคอมไพล์เลอร์ของภาษา C ในบัญชีนี้ ข้อมูลประเภท bool นั้นถูกนำออกแล้ว ดังนั้นเพื่อใช้ข้อมูลประเภทนี้ เราต้องทำเข้าไอลบรารี่จาก stdbool.h สำหรับการใช้งาน และคอมไพล์เลอร์ของคุณสนับสนุนเวอร์ชัน C99 ด้วย อย่างไรก็ตามมีหลายวิธีที่จะ Implement ข้อมูลประเภท bool ถ้าหากคุณใช้บอยๆ ในโปรแกรม หรือสามารถใช้ข้อมูลประเภท Integer แทนได้

2.3.2.4 Char (ตัวอักษร) เป็นประเภทข้อมูลที่ใช้สำหรับเก็บตัวอักษรจำนวนหนึ่งตัวอักษรใน ASCII ภาษา C นั้นยังไม่ได้สนับสนุนการเก็บข้อมูลแบบ String แต่คุณสามารถทำแบบนั้นได้ โดยการใช้ Char array เพื่อเก็บตัวอักษรที่ต้องกันหลายตัว มาดูตัวอย่างการใช้งานตัวแปรประเภท char ในภาษา C

```

#include <stdio.h>

int main()

{

    char n = 'A';

    printf("%c\n", n);

    printf("%d\n", n);

    char *name = "marcuscode";

    printf("%s", name);

    char extension[] = ".com";

    printf("%s", extension);

    return 0;
}

```

```
}
```

ในตัวอย่าง เป็นการประกาศตัวแปรประเภท char โดยการเก็บตัวอักษร 'A' ตัวอักษรทุกตัวนั้นมีรหัส ASCII ของมันในฐาน 10 ในภาษา C นั้นข้อมูลประเภท Char และ Integer สามารถแปลงค่าได้โดยวิธี implicit และ Explicit type casting ยกตัวอย่างเช่น เมื่อเราต้องการค่าที่หมายถึง A และสามารถใช้ 65 แทนได้ดังตัวอย่างข้างล่าง

```
char a = 65; // A
```

```
int b = 'A'; // 65
```

เราสามารถเก็บตัวอักษรที่มากกว่าหนึ่งตัวอักษรได้ โดยเรารายกว่าอาเรย์ หรือ Array of char ซึ่งสามารถสร้างได้ทั้งในรูปแบบของอาเรย์ปกติและพอยน์เตอร์ นี้เป็นผลลัพธ์การทำงานของโปรแกรมในการใช้งานข้อมูลประเภท Char และ Char อาเรย์

```
A
```

```
65
```

marcuscode.com

2.3.2.5 Integer (ตัวเลขจำนวนเต็ม) เป็นประเภทข้อมูลแบบจำนวนเต็ม ซึ่งข้อมูลแบบ Integer นั้นจะประกอบไปด้วยหลายขนาด เช่น short int และ long ซึ่งจะเก็บข้อมูลในระยะที่แตกต่างกัน เราสามารถใช้ตัวแปรประเภท Integer สำหรับเก็บข้อมูลที่บวกได้ เช่น จำนวนของผลไม้ในตะกร้า จำนวนของคนในห้อง จำนวนของรถที่วิ่งบนท้องถนน เป็นต้น มาดูตัวอย่างการใช้งานตัวแปรประเภท Integer ในภาษา C

```
#include <stdio.h>

int main()
{
    int apple = 3;
    int orange = 2;
    int total = apple + orange;
    printf("%d apples and %d oranges\n", apple, orange);
    printf("I have %d fruits in total", total);
    return 0;
}
```

ในตัวอย่าง เป็นการใช้งานตัวแปรประเภท Integer สำหรับเก็บจำนวนของผลไม้แต่ละชนิด และหารรวมของผลไม้ทั้งสองในตัวแปร total ต่อไปเราจะพูดเกี่ยวกับค่ามากสุดและค่าน้อยสุดของข้อมูลประเภท Integer ในภาษา C ข้างล่างนี้เป็นตัวอย่างของโปรแกรม

```
#include <stdio.h>
#include <limits.h>

int main()
{
    printf("Maximum value of int: %d\n", INT_MAX);
    printf("Minimum value of int: %d\n", INT_MIN);
    int a = INT_MAX;
```

```

    printf("%d\n", a);

    a = a + 1;

    printf("%d\n", a);

    return 0;
}

```

ในตัวอย่าง เป็นโปรแกรมในการแสดงค่ามากสุดและค่าน้อยสุดของข้อมูลประเภท Integer โดยใช้ค่าคงที่ INT_MAX และ INT_MIN ซึ่งเป็นค่าที่กำหนดโดย Marco ที่อยู่ในไฟล์ limits.h ของภาษา C และเราทดสอบการทำงานของโปรแกรมโดยการกำหนดค่ามากสุดให้กับตัวแปร a และเพิ่มค่าของตัวแปรชั้นไป ผลลัพธ์ที่ได้คือค่าในตัวแปรนั้นเกิด Overflow และเป็นที่น้อยที่สุดของ Integer ซึ่งจะได้ผลลัพธ์ตามข้างล่าง

Maximum value of int: 2147483647

Minimum value of int: -2147483648

2147483647

-2147483648

2.3.2.6 Floating-point numbers (ตัวเลขจำนวนจริง) เป็นประเภทข้อมูลสำหรับเก็บตัวเลขจำนวนจริง ในภาษา C จะมีอยู่สองประเภทคือ float และ double สิ่งที่แตกต่างกันคือขนาดของพื้นที่ที่ใช้ในการจัดเก็บ และข้อมูลประเภท double นั้นจะเก็บจำนวนของหน่วยนิยมได้มากกว่า ข้อมูลประเภทนี้จะใช้เก็บข้อมูลที่มีค่าเป็นเศษส่วนหรือตัวเลขที่มีเลขหลังจุดหน่วยนิยม ยกตัวอย่างเช่น ข้อมูลการคำนวณทางวิทยาศาสตร์ เป็นต้น มาดูตัวอย่างการประยุกต์ใช้แบบใช้งาน

```

#include <stdio.h>

int main()
{
    float pi = 3.14f,
          half = 3 / 2.0f;
    float half2 = 3 / 2;
    printf("Pi: %f\n", pi);
    printf("Half of 3: %f\n", half);
    printf("Half of 3: %f\n", half2);
    double gravity = 9.807;
    double earth_mass = 5.972E24;
    double meter = 1E-3;
    printf("Earth gravity: %lf m/s^2\n", gravity);
    printf("Earth mass: %lf kg\n", earth_mass);
    printf("1 centimeter: %lf meter\n", meter);
    return 0;
}

```

ในตัวอย่าง เป็นการใช้งานตัวแปรสำหรับเก็บข้อมูลในรูปแบบของทศนิยมทั้งตัวแปรประเภท float และ double ในส่วนแรกเราได้ประกาศตัวแปร float จำนวนสามตัวและกำหนดค่าให้กับตัวแปร ค่าของข้อมูลประเภท float นั้นจะลงทะเบียนด้วย f หรือ F เมื่อ

```
float half = 3 / 2.0f;
```

```
float half2 = 3 / 2;
```

ต่อมาเป็นการทดสอบการหารทั่วเลขในตัวแปร half และ half2 เพื่อให้คุณเข้าใจและเห็นความแตกต่างของสิ่งที่เกิดขึ้น การคูณหรือการหารทั่วเลขในภาษา C นั้นจะได้ผลลัพธ์เป็น Integer ถ้าหากตัวเลขทั้งสองตัวนั้นเป็นประเภท Integer เมื่อในตัวแปร half2 ถ้าหากมีตัวแปรอย่างน้อยหนึ่งตัวแปรที่เป็น float หรือ double จะได้ข้อมูลเป็น float หรือ double นี้จึงเป็นสิ่งที่ต้องควรระวังในการเขียนโปรแกรม

```
double gravity = 9.807;
```

```
double earth_mass = 5.972E24;
```

```
double meter = 1E-3;
```

สามารถตัวแปรต่อมาเป็นการประกาศตัวแปรประเภท double สำหรับตัวแปรทั้งสองประเภทนั้นสามารถประกาศในรูปแบบสัญกรณ์วิทยาศาสตร์ได้ เช่น ในตัวอย่างในตัวแปร earth_mass เป็นการกำหนดค่าแบบย่อซึ่งมีค่าเท่ากับ 5.972×10^{24} และในตัวแปร meter นั้นมีค่าเท่ากับ 1

```
 $\times 10^{-3}$ 
```

```
Pi: 3.140000
```

```
Half of 3: 1.500000
```

```
Half of 3: 1.000000
```

```
Earth gravity: 9.807000 m/s^2
```

```
Earth mass: 5972000000000000300000000.000000 kg
```

```
1 centimeter: 0.001000 meter
```

นี่เป็นผลลัพธ์ของโปรแกรมในการทำงานกับประเภทข้อมูลแบบทศนิยมในภาษา C ด้วยตัวแปรแบบ float และ double

2.3.3 ตัวดำเนินการ

ตัวดำเนินการจะถูกใช้กับตัวแปรและค่าคงที่ในการดำเนินการทางคณิตศาสตร์ ในภาษา C มีตัวดำเนินการประเภทต่างๆ ที่ทำหน้าที่แตกต่างกันไป ในบทนี้ คุณจะได้เรียนรู้เกี่ยวกับตัวดำเนินการและตัวอย่างการใช้ตัวดำเนินการประเภทต่างๆ ในภาษา C

ข้างล่างนี้เป็นรายการของตัวดำเนินการในภาษา C

- Assignment operator
- Arithmetic operators
- Compound assignment
- Increment and decrement
- Relational and comparison operators
- Logical operators
- Bitwise operators
- Operator precedence

2.3.3.1 Assignment operator ตัวดำเนินการกำหนดค่า ในภาษา C ใช้สัญลักษณ์เท่ากับ (=) มันถูกใช้เพื่อกำหนดค่าให้กับตัวแปรหรือค่าคงที่ ตัวดำเนินการจะมีสอง Operand การทำงานของมันคือการนำค่าทางด้านขวาไปใส่ทางด้านซ้าย และ Operand ทางด้านขวาสามารถเป็น Expression ได้ๆ

```
a = 2;      b = 3;
```

```
c = a + b;
```

ในตัวอย่างนี้ เราได้สร้างสามตัวแปรและกำหนดค่า 2 ให้กับตัวแปร a กำหนดค่า 3 ให้กับตัวแปร b และผลรวมของ a และ b ให้กับตัวแปร c ตามลำดับ

2.3.3.2 Arithmetic operators (+, -, *, /, %) ตัวดำเนินการทางคณิตศาสตร์ คือตัวดำเนินการที่ใช้เพื่อกระทำการดำเนินการทางคณิตศาสตร์ระหว่างตัวแปรหรือค่าคงที่ เช่น การบวก การลบ การคูณ และการหาร สำหรับในการเขียนโปรแกรมในภาษา C นั้นจะมีตัวดำเนินการสำหรับการหารเอาเศษ (Modulo) เพิ่มเข้ามา

ตารางข้างล่างนี้คือตัวดำเนินการทางคณิตศาสตร์ในภาษา C

Symbol	Name	Example
+	Addition	c = a + b
-	Subtraction	c = a - b
*	Multiplication	c = a * b
/	Division	c = a / b
%	Modulo	c = a % b

จากในตารางของตัวดำเนินการข้างบน คุณน่าจะคุ้นเคยกับมันมากบ้างแล้วในการเรียนในวิชาคณิตศาสตร์ นอกจากตัวดำเนินการ Modulo ที่เป็นการหารเอาเศษ ต่อไปมาดูตัวอย่างการใช้ตัวดำเนินการทางคณิตศาสตร์ ในการเขียนโปรแกรมในภาษา C

```
#include <stdio.h>
int main()
{
    int a = 5;
    int b = 10;
    printf("a + b = %d\n", a + b);
    printf("a - b = %d\n", a - b);
    printf("a * b = %d\n", a * b);
```

```

printf("a / b = %d\n", a / b);
printf("a % b = %d\n", a % b);
return 0;
}

```

ในตัวอย่าง เป็นการใช้งานตัวดำเนินการประเพณีต่างๆ ในภาษา C เราได้ประกาศตัวแปร a และ b และพร้อมกับกำหนดค่าให้กับมันและใช้ตัวดำเนินการต่างๆ ซึ่งคุณคงจะคุ้นเคยดี และในตัวดำเนินการ Modulo นั้นเป็นการหารเอาเศษซึ่งคำตอบที่ได้จะเป็นเศษของการหาร เมื่อรันโปรแกรม มันจะได้ผลลัพธ์ดังนี้

a + b = 15

a - b = -5

a * b = 50

a / b = 0

a % b = 5

2.3.3.3 Compound assignment ($+=$, $-=$, $*=$, $/=$, $%=$, $>>=$, $<<=$, $&=$, $\wedge=$, $|=$) คือตัวดำเนินการที่ใช้เพื่ออัพเดทหรือแก้ไขค่าปัจจุบันของตัวแปรโดยการกระทำการทางคณิตศาสตร์และใช้ตัวดำเนินการกำหนดค่าร่วมด้วย ซึ่งตัวดำเนินการแบบ Compound assignment มักจะใช้เป็นรูปแบบสั้นของตัวดำเนินการทางคณิตศาสตร์และตัวดำเนินการระดับบิตเพื่อให้การเขียนสั้นลง

ข้างล่างนี้เป็นตารางของ Compound assignment operators ในภาษา C

Operator	Example	Equivalent to
$+=$	$a += 2;$	$a = a + 2$
$-=$	$a -= 2;$	$a = a - 2$
$*=$	$a *= 2;$	$a = a * 2$
$/=$	$a /= 2;$	$a = a / 2$
$%=$	$a %= 2;$	$a = a \% 2$
$>>=$	$a >>= 2;$	$a = a >> 2$
$<<=$	$a <<= 2$	$a = a << 2$

&=	a &= 2;	a = a & 2
----	---------	-----------

^=	a ^= 2;	a = a ^ 2
----	---------	-----------

=	a = 2;	a = a 2
---	---------	-----------

มาดูตัวอย่างของการใช้งานของตัวดำเนินการ Compound assignment

```
#include <stdio.h>

int main()
{
    int x = 10;
    int y = 2;
    x += 10; //equivalent to x = x + 10
    y -= 2; //equivalent to y = y - 2
    printf("x = %d, y = %d", x, y);
    return 0;
}
```

ในตัวอย่าง นั้นเป็นรูปแบบอย่างสั้นในการดำเนินการทางคณิตศาสตร์ของมัน โดยการทำางานของตัวดำเนินการจะอ้างอิงจากค่าเดิม เช่น ในตัวแปร x นั้นเป็นการบวกค่าเข้าไปในตัวแปรเดิมอีก 10 และในตัวแปร y นั้นเป็นการลบค่าออกจากตัวแปรเดิมออก 2 และมันจะได้ผลลัพธ์ดังนี้

x = 20, y = 0

2.3.3.4 Increment and decrement (++, --) ตัวดำเนินการเพิ่มและลดค่า คือตัวดำเนินการที่ใช้เพื่อบวกหรือลบค่าออกจากตัวแปรโดย 1 โดยการเพิ่มเครื่องหมาย ++ หรือ-- ใส่ข้างหน้าหรือข้างหลังตัวแปร ซึ่งมีรูปแบบการใช้ดังนี้
identifier++

มาดูตัวอย่างของการใช้ตัวดำเนินการเพิ่มและลดค่า

```
#include <stdio.h>

int main()
{
    int a = 1;
    int b = 10;
    a++;
    b--;
    printf("a=%d, b=%d", a, b);
    return 0;
}
```

```
}
```

เมื่อเรารันโปรแกรม นี้เป็นผลลัพธ์ที่เราจะได้

```
a=2, b=9
```

เหตุ因ที่คุณเห็นในตัวอย่าง มันเป็นรูปแบบอย่างสั้นของ $a = a + 1$ และ $b = b - 1$ มันมักจะใช้กับคำสั่ง for loop เพื่อเพิ่มของการรัน Index หรือตำแหน่ง Index ของอาเรย์

และยังมีรูปแบบอื่นของตัวดำเนินการนี้คือ Prefix เช่น $++a$ $--b$ โดยมันหมายถึงจะมีการเพิ่มหรือลดค่าก่อนที่จะมีการประมวลผลคำสั่งปัจจุบัน

2.3.3.5 Relational และ comparison operators ($==$, $!=$, $>$, $<$, \geq , \leq) ตัวดำเนินการความสัมพันธ์และเปรียบเทียบ คือตัวดำเนินการที่ถูกใช้เพื่อประเมินค่า true และ false ระหว่างสองค่าถูกดำเนินการ ซึ่งขึ้นกับเงื่อนไขและความสัมพันธ์ของมัน ข้างล่างนี้เป็นรายการของตัวดำเนินการ Relational และตัวดำเนินการเปรียบเทียบ

Operator	Example	Result
$==$	$a == b$	true if 'a' equal to 'b', otherwise false
$!=$	$a != b$	true if 'a' not equal to 'b', otherwise false
$<$	$a < b$	true if 'a' less than 'b', otherwise false
$>$	$a > b$	true if 'a' greater than 'b', otherwise false
\leq	$a \leq b$	true if 'a' less than or equal to 'b', otherwise false
\geq	$a \geq b$	true if 'a' greater than or equal to 'b', otherwise false

ตัวดำเนินการเหล่านี้ถูกใช้ เช่น เพื่อเปรียบเทียบค่าระหว่างตัวแปรว่าเท่ากันหรือไม่ หรือเปรียบการมากกว่าน้อยกว่าหรือค่าในตัวแปร และผลลัพธ์ของการเปรียบเทียบจะเป็น true และ false

```
#include <stdio.h>
int main()
{
    int a = 5;
    int b = 10;
    if (a == b)
    {
```

```

    printf ("a and b are equal");

}

else { printf ("a and b are not equal"); }

return 0;

}

```

ในตัวอย่าง เป็นโปรแกรมในการเปรียบเทียบค่าในตัวแปร a และ b โดยการใช้คำสั่ง if ซึ่งถ้าหากค่าของตัวแปรทั้งสองเท่ากันซึ่งจะทำให้เงื่อนไขเป็นจริง จะทำให้โปรแกรมทำงานในบล็อกของคำสั่ง if "ไม่เข่นนั้นโปรแกรมจะทำในบล็อกของคำสั่ง else แทน

a and b are not equal

นี้เป็นผลลัพธ์การทำงานของโปรแกรม จะเห็นว่าโปรแกรมทำงานในบล็อกของคำสั่ง else เพราะว่าค่าในตัวแปรทั้งสองนั้นไม่เท่ากัน ให้คุณลองเปลี่ยนค่าในตัวแปรให้เท่ากันแล้วลองรันโปรแกรมอีกครั้งเพื่อดูผลลัพธ์

2.3.3.6 Logical operators (!, &&, ||) ตัวดำเนินการทางตรรกศาสตร์ถูกใช้เพื่อประเมิน Expression ย่อยหลายๆ Expression ให้เหลือเพียงอันเดียว โดยผลลัพธ์สุดท้ายนั้นจะเป็นจริงหรือเท็จ ข้างล่างนี้เป็นรายการตัวดำเนินการทางตรรกศาสตร์ในภาษา C

Name	Symbol	Example
not	!	!a
and	&&	a && b
or		a b

ในการทำงานของตัวดำเนินการนั้น ตัวดำเนินการ ! (not) จะกลับค่าของ Boolean expression ตัวดำเนินการ && (and) ทำการเชื่อมสอง Expression เข้าด้วยกัน โดยจะได้ผลลัพธ์เป็นจริงหากทั้งสองค่าเป็นจริง "ไม่เข่นนั้นจะเป็นเท็จ และตัวดำเนินการ || (or) เชื่อมสอง Expression เข้าด้วยกัน โดยจะได้ผลลัพธ์เป็นจริงหากมีอย่างน้อยหนึ่ง Expression ที่เป็นจริง "ไม่เข่นนั้นจะเป็นเท็จ ต่อไปมาดูตัวอย่างการใช้งานของตัวดำเนินการ ตรรกศาสตร์

```

#include <stdio.h>

int main()
{
    int a = 10;
    bool b = 1;
    bool c = 0;
    // using "not" operator
    if (!(a == 10))

        printf("a is not equal to 10.\n");

```

```

else
    printf("a is equal to 10.\n");
// using "and" operator
if (b && c)
    printf("True.\n");
else
    printf("False.\n");
// using "or" operator
if (b || c)
    printf("True.\n");
else
    printf("False.\n");
return 0;
}

```

ในตัวอย่างเรามีตัวแปรสามตัว ตัวแปร a เป็น integer ในขณะที่ตัวแปร b และ c เป็น boolean ในภาษา C ค่าของ boolean ที่เป็นจริงนั้นจะมีค่าเป็น 1 และเท็จจะเป็น 0

และนี่คือผลลัพธ์ของโปรแกรม

a is equal to 10.

False.

True.

2.3.3.7 Bitwise operators (`&`, `|`, `~`, `<<`, `>>`) นั้นถูกใช้ในการดำเนินการระดับบิตของตัวแปรหรือข้อมูล มันมีประโยชน์มากในการเขียนโปรแกรมระดับต่ำ ตัวดำเนินการ Bitwise นั้นใช้หลักการทำงานเหมือนกับตัวดำเนินการทางตรรกศาสตร์ โดยใช้ 1 สำหรับค่าจริงและ 0 สำหรับค่าเท็จ นี่เป็นรายการของตัวดำเนินการ Bitwise ในภาษา C

Symbol	Name	Description
<code>&</code>	Bitwise AND	1 ถ้าบิตทั้งคู่เป็น 1, ไม่เช่นนั้นเป็น 0
<code> </code>	Bitwise inclusive OR	1 ถ้าอย่างน้อยหนึ่งบิตเป็น 1, ไม่เช่นนั้นเป็น 0
<code>^</code>	Bitwise exclusive OR	1 ถ้าทั้งสองบิตแตกต่างกัน, ไม่เช่นนั้นเป็น 0
<code>~</code>	bit inversion	กลับบิตจาก 1 เป็น 0 และในทางตรงข้าม

<<	Shift bits left	เลื่อนบิตไปทางซ้าย เดิมบิต 0 ทางขวา
----	-----------------	-------------------------------------

>>	Shift bits right	เลื่อนบิตไปทางขวา เดิมบิต 0 ทางซ้าย
----	------------------	-------------------------------------

จากตารางข้างบน เป็นตัวดำเนินการประเพณีต่างๆ ในภาษา C ที่ใช้ดำเนินการกับข้อมูลในระดับบิต ยกตัวอย่าง เช่น เมื่อเรากำหนดค่าให้กับตัวแปรนั้น คอมพิวเตอร์จะเก็บข้อมูลในรูปแบบของเลขฐานสอง (Binary) ตัวดำเนินการเหล่านี้ใช้สำหรับจัดการกับข้อมูลตั้งกล่าว มาดูตัวอย่างการใช้งานในภาษา C

```
#include <stdio.h>

int main()
{
    int a = 2; // 00000010
    int b = 5; // 00000101
    printf("a & b = %d\n", a & b); // 00000000 = 0
    printf("a & b = %d\n", a | b); // 00000111 = 7
    printf("~a = %d\n", ~a); // 11111101 = -1
    printf("a >> 1 = %d\n", a >> 1); // 00000001 = 1
    printf("a << 1 = %d\n", a << 1); // 00000100 = 8
    return 0;
}
```

ในตัวอย่าง เป็นโปรแกรมในการใช้งานตัวดำเนินการระดับบิตกับการจัดการข้อมูลประเภท Integer เราได้ประกาศตัวแปร a และ b และได้ค่ามnen ตัวของ Binary ที่ถูกเก็บไว้ในหน่วยความจำ การทำงานของตัวดำเนินการ & และ | นั้นทำงานกับคู่ของแต่ละบิตของตัวแปรทั้งสอง ส่วนอีกสามตัวดำเนินการที่เหลือนั้นจะทำการกับตัวแปรเดียว ในตัวดำเนินการ ~ นั้นเป็นการกลับบิต และสำหรับตัวดำเนินการ Bit shift นั้นเมื่อเลื่อนบิตไปทางซ้ายจะทำให้ค่าเพิ่มขึ้นสองเท่า และเมื่อเลื่อนบิตไปทางขวาจะทำให้ค่าลดลงสองเท่า

```
a & b = 0
a & b = 7
~a = -3
a >> 1 = 1
a << 1 = 4
```

นี่เป็นผลลัพธ์การทำงานของโปรแกรมของการใช้งานตัวดำเนินการระดับบิตในภาษา C

2.3.3.8 Operator precedence ลำดับความสำคัญของตัวดำเนินการ ใช้เพื่อกำหนดว่าตัวดำเนินการใช้ที่จะถูกทำงานก่อน ยกตัวอย่าง เช่น ในการทำงานของ Expression $6 + 3 * 4$ นั้นจะได้ผลลัพธ์เป็น 18 เพราะว่าตัวดำเนินการ * มีความสำคัญมากกว่าตัวดำเนินการ + คุณสามารถบังคับให้การบวกทำงานก่อนได้โดยใช้วงเล็บ $(6 + 3) * 4$ เพื่อให้การบวกสามารถทำงานก่อนได้ ซึ่งจะได้ผลลัพธ์เป็น 36 เป็นต้น ข้างล่างนี้ เป็นตารางแสดงลำดับความสำคัญของตัวดำเนินการในภาษา C

Operator Name	Associativity	Operators
Primary	left to right	()[].
Unary	ขวาไปซ้าย	+ - -! ~ & * (type_name) sizeof new delete
Multiplicative	ซ้ายไปขวา	* / %
Additive	ซ้ายไปขวา	+
Bitwise Shift	ซ้ายไปขวา	<<>>
Relational	ซ้ายไปขวา	<><=>=
Equality	ซ้ายไปขวา	== !=
Bitwise AND	ซ้ายไปขวา	&
Bitwise Exclusive OR	ซ้ายไปขวา	^
Bitwise Inclusive OR	ซ้ายไปขวา	
Logical AND	ซ้ายไปขวา	&&
Logical OR	ซ้ายไปขวา	
Conditional	ขวาไปซ้าย	? :
Assignment	ขวาไปซ้าย	= += -= *= /= <<=>>= %= &= ^= =
Comma	ซ้ายไปขวา	,

ในบทนี้ เราได้ครอบคลุมการดำเนินการพื้นฐานในภาษา C คุณได้เรียนรู้และทำความเข้าใจในตัวดำเนินการประเภทต่างๆ และได้เห็นถึงตัวอย่างการใช้งานของตัวดำเนินการแต่ละรูปแบบ ซึ่งจะนำไปใช้ในการเขียนโปรแกรมในบทต่อๆ ไป

2.3.4 อินพุต เอ้าต์พุต

2.3.4.1 Input-output กับประเภทข้อมูลในภาษา C มีฟังก์ชันให้เราสามารถรับค่าและแสดงผลได้ โดยจะมีรูปแบบในการรับค่าและแสดงผลของข้อมูลขึ้นกับประเภทของข้อมูล นี่เป็นข้อมูลส่วนมากที่เราจะพบในการทำงาน โดยจะเป็น specifier ที่จะใช้ในการรับข้อมูลแต่ละแบบ

specifier	Description	Usage
d	Integer	ใช้กับตัวเลขจำนวนเต็ม
f	Floating, double	ใช้กับตัวเลขจำนวนจริง
c	Character	ใช้กับตัวอักษรหนึ่งตัวอักษร
s	String of characters	ใช้กับสตริงหรือตัวอักษรหลายตัว
p	Pointer	ใช้กับค่าที่อยู่ของหน่วยความจำ
%	%	เครื่องหมายที่จะตามโดย specifier ตัวอื่นๆ

2.3.4.2 ฟังก์ชัน printf ในการแสดงผลในภาษา C ฟังก์ชันที่ใช้คือฟังก์ชัน printf ซึ่งเป็นฟังก์ชันที่สามารถแสดงข้อมูลประเภทต่างๆ ทางหน้าจอภาพได้ โดยมันมีรูปแบบการใช้งานดังนี้

```
printf("string pattern", value1, value2, ...);
```

ในการใช้ฟังก์ชัน อาร์กิวเม้นต์ตัวแรกคือรูปแบบการแสดงผล และอาร์กิวเม้นต์ตัวต่อไปเป็นข้อมูลที่ตรงกับ specifier ที่เรากำหนดไว้ตามลำดับ มาดูตัวอย่างการใช้งานจริง

```
printf("Hello, my name is Mateo\n");
int age = 20;
printf("\nI'm %d year old\n", age);
printf("I have %f USD and %d cents\n", 10.5, 4);
```

จากตัวอย่างเป็นการใช้ฟังก์ชัน printf เพื่อการแสดงผลแบบต่างๆ อันแรกเป็นการแสดงผลข้อความโดยที่ไม่มี specifier อันที่สองเราต้องการแสดงอายุที่เป็น Integer เราจึงใช้ %d เพื่อแทนการแสดงผลของมัน อันสุดท้ายแสดงให้เห็นถึงการแสดงผลหลายค่าโดยเราต้องกำหนด specifier ให้ตรงกับค่าของเรา

และนี่เป็นตัวอย่างผลลัพธ์ของโปรแกรม

```
Hello, my name is Mateo
```

```
I'm 20 years old
```

```
I have 4.5 USD and 4 cents
```

2.3.4.3 พิ้งก์ชัน scanf ในการรับค่านั้นเราจะใช้พิ้งก์ชัน scanf ใน การรับค่า โดยการรับค่าเรามักจะหมายถึงการรับค่าจากคีย์บอร์ด โดยรูปแบบการใช้งานของพิ้งก์ชันเป็นดังนี้

```
scanf("%specifier1 %specifier2 ...", variable1, variable2, ...);
```

พิ้งก์ชัน scanf นั้น มีรูปแบบที่คล้ายกันกับพิ้งก์ชัน printf โดย อาร์กิวเมนต์ในพิ้งก์ตัวแรกเป็นสตริงของ specifier input ที่เราต้องการรับ และเป็นลำดับของตัวแปรที่ตรงกับ specifier ที่เราได้กำหนดไว้ในอาร์กิวเมนต์ตัวแรก และใส่ตัวแปรให้ตรงกับลำดับของ specifier ที่กำหนด

ต่อไปมาดูตัวอย่างของการรับค่าโดยการใช้พิ้งก์ชัน scanf

```
int a;
float b;
scanf("%d%f", &a, &b);
printf("value of a is %d\n", a);
printf("value of b is %f\n", b);
```

ในตัวอย่าง เราได้มีตัวแปรสองตัวคือ a และ b และมีประเภทของตัวแปรเป็น integer และ floating ตามลำดับ ดังนั้นใน specifier input เราจึงใส่ %d สำหรับตัวแปร a และ %f สำหรับตัวแปร b อาร์กิวเมนต์ตัวอื่นเราจะใส่ลำดับของตัวแปร โดยลำดับของตัวแปรจะแตกต่างจากพิ้งก์ชัน printf โดยจะนำหน้าด้วยเครื่องหมาย & ซึ่งมันหมายถึงที่อยู่ของตัวแปร ถ้าในกรณีที่ตัวแปรเป็นอาร์ย์ของตัวอักษรนั้นเราไม่ต้องใส่

และนี่เป็นตัวอย่างของโปรแกรมเมื่อใส่ค่าเป็น 5 และ 10.5 ตามลำดับ

```
5 10.5
```

```
value of a is 5
```

```
value of b is 10.500000
```

2.3.4.4 Escape characters เป็นตัวอักษรพิเศษในภาษา C ที่จะต้องใช้สัญลักษณ์เครื่องหมาย \ นำหน้าเสมอ เพื่อไม่ให้เกิดข้อผิดพลาดในการคอมไพล์โปรแกรม เพราะว่าตัวอักษรนี้เป็นสัญลักษณ์ที่ตรงกันกับโครงสร้างของภาษา นี่เป็นรายการของตัวอักษรพิเศษภาษา C

Escape character	Character represented
\a	Alarm
\b	Backspace

\n	Newline
\r	Carriage Return
\t	Horizontal Tab
\v	Vertical Tab
\\\	Backslash
\'	Single quotation mark
\"	Double quotation mark
\?	Question mark

ต่อไปมาดูตัวอย่างการใช้งานของ Escape characters ในภาษา C ซึ่งเราจะเก็บข้อความที่มีตัวอักษรเหล่านี้ไว้ภายในตัวแปร char และตัวแปร char array

```
#include <stdio.h>

int main ()
{
    char tab = '\t';
    char a = 'A';
    char text[] = "His name is \"Mateo\".\n";
    char text2[] =
        "You can visit Mateo's website by asking him for address.\n";
    printf("%c%c\n", tab, a);
    printf("%s", text);
    printf("%s", text2);
    return 0;
}
```

ในตัวอย่าง เป็นการใช้งาน Escape characters ในการเขียนโปรแกรม ในสองตัวแปรแรก tab และ a นั้นเป็นตัวแปรแบบ char ตัวแปรแรกเป็นตัวแปรที่เป็นตัวอักษรพิเศษซึ่งหมายถึง tab และตัวแปรที่สองเป็นค่าปกติของ char ที่ไม่ใช่ตัวอักษรพิเศษ

ตัวมาในตัวแปร text และ text2 เป็นตัวแปรแบบ char array หรือตัวแปรที่เก็บข้อมูลของ char ได้มากกว่า 1 ตัวอักษร สำหรับการกำหนดค่าให้กับตัวแปรเราจึงใช้ \" สำหรับเครื่องหมาย \" และ \' สำหรับเครื่องหมาย ' ส่วน \g นั้นเป็นตัวอักษรพิเศษที่หมายถึงการขับบรรทัดใหม่ซึ่งเราจะใช้บ่อยในบทเรียนนี้

A

His name is "Mateo".

You can visit Mateo's website by asking him for address.

นี่เป็นผลลัพธ์การทำงานของโปรแกรมในการใช้งาน Escape characters ในภาษา C

ในบทนี้ คุณได้เรียนรู้เกี่ยวกับการรับค่าและการแสดงผลในภาษา C เบื้องต้น เพื่อให้สามารถสร้างโปรแกรมที่มีการติดต่อกับผู้ใช้ได้ อย่างไรก็ตาม Input-output นั้นสามารถเป็นได้ในทางอื่นๆ เช่น การรับค่าจากเสียง กล้องถ่ายรูป หรืออาจจะเป็นการทำงานกับไฟล์ หรือแม้กระทั่งกับ Network สำหรับบทนี้ เราพูดถึงการรับค่าทางคีย์บอร์ดและการแสดงผลทางหน้าจอ Console

2.3.5 คำสั่งควบคุม

2.3.5.1 คำสั่ง If คำสั่งควบคุมการทำงานที่เป็นพื้นฐานที่สุดในภาษา C นั้นก็คือคำสั่ง if มันใช้สำหรับสร้างเงื่อนไขให้โปรแกรมทำงานตามที่ต้องการ ยกตัวอย่างเช่น ถ้าหากวันนี้ฝนไม่ตก คุณจะออกไปเที่ยวข้างนอก นี่เป็นการตัดสินใจที่เกิดขึ้นและถูกนำมาระบุคิดมาใช้ในการเขียนโปรแกรม นี่เป็นรูปแบบของการใช้งานคำสั่ง if ในภาษา C

```
if (expression)
{
    // statements
}
```

คำสั่ง if เป็นคำสั่งที่ใช้ในการตรวจสอบว่าเงื่อนไขเป็นจริงหรือไม่ โดยการใช้ตัวดำเนินการต่างๆ ในการสร้างเงื่อนไขหรือ expression ถ้าผลลัพธ์เป็นจริง โปรแกรมจะทำงานในบล็อกของคำสั่ง if ที่เราได้กำหนดขึ้น และถ้าหากผลลัพธ์ไม่เป็นจริงโปรแกรมจะข้ามการทำงานในบล็อกคำสั่งไป มาดูตัวอย่างการใช้งานของมัน;

```
#include <stdio.h>

int main()
{
    if (10 == 10)
    {
        printf("block 1 is executed.\n");
    }
    if (3 < 1)
    {
```

```

    printf("block 2 is executed.\n");

}

int a = 10;

if (a % 2 == 0)

{

    printf("block 3 is executed.\n");

}

return 0;
}

```

ในการทำงานของโปรแกรมเป็นการใช้คำสั่ง if เพื่อเปรียบเทือนไข่ต่างๆ โดยแต่ละบล็อกคำสั่งจะทำงานเมื่อเงื่อนไขเป็นจริงเท่านั้น บล็อกแรกจะทำงาน เพราะว่า 10 มีค่าเท่ากับ 10 บล็อกที่สองจะไม่ทำงาน เพราะว่า 3 ไม่น้อยกว่า 1 และสำหรับบล็อกสุดท้ายจะทำงาน เพราะว่า a เป็นจำนวนคู่

และนี้เป็นผลลัพธ์ของโปรแกรมเมื่อรัน คุณจะเห็นว่ามีแค่ block 1 และ 3 ที่แสดงผล เพราะว่าเงื่อนไขของ if เป็นจริง

```

block 1 is executed

block 3 is executed

```

2.3.5.2 คำสั่ง if else เป็นคำสั่งมีการเพิ่มเงื่อนไข else if เข้ามาสำหรับการสร้างเงื่อนไขแบบหลายทางเลือก นั่นหมายถึงในบล็อกของคำสั่งจะทำงานเมื่อเงื่อนไขไม่ต้องกับเงื่อนไขก่อนหน้าแต่ตรงกับเงื่อนไขปัจจุบัน นอกจากนี้ยังมีคำสั่ง else ซึ่งเป็นบล็อกของคำสั่งที่จะทำงานถ้าหากไม่มีเงื่อนไขใดเป็นจริงเลย มาดูตัวอย่างของการใช้คำสั่ง if-else

```

#include <stdio.h>

int main()

{

    int score;

    scanf ("%d", &score);

    if (score < 50)

        printf("Your grade is F");

    else if (score < 60)

        printf("Your grade is D");

    else if (score < 70)

        printf("Your grade is C");

    else if (score < 80)

        printf("Your grade is B");

    else

        printf("Congratulation! You got grade A");

    return 0;
}

```

ในตัวอย่างนี้เป็นโปรแกรมสำหรับการคำนวณเกรด โดยโปรแกรมจะให้คุณสามารถใส่เกรดเข้ามา และเปรียบเทียบเงื่อนไขโดยใช้คะแนนตรวจสอบ คุณจะสังเกตเห็นว่าเราสามารถ ใช้ else if ได้ถ้าหากมีหลายเงื่อนไข และใช้คำสั่ง else เป็นเงื่อนไข default ในกรณีที่ไม่ตรงกับเงื่อนไขใดๆ เลย

74

Your grade is B

85

Congratulation! You got grade A

นี่เป็นผลลัพธ์การทำงานของโปรแกรมเมื่อเราได้กรอกคะแนนเป็น 74 ซึ่งตรงกับเงื่อนไขของเกรด B หลังจากโปรแกรมพบรอย่างน้อยหนึ่งเงื่อนไขที่ เป็นจริง บล็อกที่เหลือจะถูกข้ามไปในทันที และเราได้รับโปรแกรมอีกครั้งและกรอกคะแนนเป็น 85 ซึ่งโปรแกรมไม่ตรงกับเงื่อนไขใดเลย ใน บล็อกของคำสั่ง if และ else if ทำให้ในบล็อกของคำสั่ง else ทำงานและได้เกรด A

2.3.5.3 Switch case เป็นคำสั่งเงื่อนไขที่ทำงานคล้ายกับ if แต่ส่วนมากจะใช้สำหรับการเปรียบเทียบกับค่าคงที่และต้องเป็นข้อมูล ประเภท Integer หรือ Char ท่านั้น มาดูตัวอย่างของการใช้คำสั่ง Switch

```
#include <stdio.h>

int main()
{
    int number = 5;
    switch (number)
    {
        case 1:
            printf("n is 1");
            break;
        case 2:
            printf("n is 2");
            break;
        case 3:
            printf("n is 3");
            break;
        default:
            printf("Unknown");
    }
    return 0;
}
```

ในการทำงานจากตัวแปรด้านบน เป็นการส่งตัวแปร number โดยจะนำไปเปรียบเทียบกับแต่ละ case ถ้าเงื่อนไขตรง โปรแกรมจะทำงานทันที จนกว่าจะสิ้นสุดบล็อกของคำสั่ง switch ดังนั้นเราจำเป็นต้องใช้คำสั่ง break เพื่อให้จบการทำงาน สำหรับคำสั่ง default จะเป็นการทำหลังจากที่ number ไม่ตรงกับเงื่อนไขใดๆ ก่อนหน้า

2.3.5.4 while loop คำสั่งวนซ้ำที่ง่ายที่สุดคือคำสั่ง while loop โดยการทำงานของคำสั่ง while loop จะทำงานเมื่อเงื่อนไขเป็นจริง และจะสิ้นสุดการทำงานจนกว่าเงื่อนไขจะเป็นเท็จ มาดูตัวอย่างของโปรแกรม

```
#include <stdio.h>

int main()
{
    int n = 1;

    while (n <= 10)
    {
        printf("%d, ", n);

        n++;
    }

    return 0;
}
```

ในตัวอย่างเป็นการใช้คำสั่ง while เพื่อแสดงผลตัวเลข 1 ถึง 10 โดยเราได้กำหนดค่าเริ่มต้นให้ n เท่ากับ 1 และจะเพิ่มขึ้นในการแสดงผลแต่ละครั้ง จนกว่าจะมากกว่า 10 จึงสิ้นสุดการทำงานของโปรแกรม และจะได้ผลลัพธ์ดังนี้

1, 2, 3, 4, 5, 6, 7, 8, 9, 10,

2.3.5.5 คำสั่ง do while นั้นคล้ายกับคำสั่ง while แต่สิ่งที่แตกต่างกันเล็กน้อยคือ คำสั่ง do while จะทำงานก่อนอย่างน้อยหนึ่งรอบ และจะตรวจสอบและถ้าเงื่อนไขไม่เป็นจริงจะสิ้นสุดการทำงาน มาดูตัวอย่างการใช้งาน

```
#include <stdio.h>

int main()
{
    int n = 1;

    do
    {
        printf("Enter number: ");

        scanf ("%d", &n);

        printf("You have entered %d\n", n);
    }

    while (n != 0);

    printf("The program is terminated");

    return 0;
}
```

ในตัวอย่างนี้ เป็นโปรแกรมที่จะรับค่าจากคีย์บอร์ด โดยจะรับตัวเลขเข้ามาและแสดงผลตัวเลขที่ได้รับเข้ามา และจะสิ้นสุดการทำงานเมื่อตัวเลขเป็น

```

1
You have entered 1

3
You have entered 3

0
You have entered 0

The program is terminated

```

2.3.5.6 คำสั่ง for loop เป็นคำสั่งวนซ้ำที่เราสามารถกำหนดค่าเริ่มต้นเงื่อนไขและการเปลี่ยนแปลงของค่าได้ ในตอนแรกครั้งเดียว โดยมีรูปแบบดังนี้

```

for (initial; expression; increment)

{
    // statements
}

```

คำสั่ง for loop สามารถใช้ได้เหมือนกับ คำสั่ง while loop และ do while loop แต่โดยทั่วไปแล้ว เรามักจะใช้คำสั่ง for loop กับการวนซ้ำในจำนวนรอบที่แน่นอน หรือมันมักจะถูกใช้กับอาเรย์ มาดูตัวอย่างการใช้งาน

```

int i;

for (i = 0; i <= 10; i++) {
    printf("%d, ", i);
}

```

ในตัวอย่างเป็นการใช้คำสั่ง for loop เพื่อวนซ้ำแสดงผลลัพธ์เลข 1 ถึง 10 เมื่อเทียบกับตัวอย่างของคำสั่ง while loop ก่อนหน้า มาดูตัวอย่างเพิ่มเติมในการใช้คำสั่ง for loop

```

#include <stdio.h>

int main()

{
    char n[] = "Hello World";

    int i;

    for (i = 0; i < strlen(n); i++)
    {
        printf("%c", n[i]);
    }

    printf("\n");

    for (i = strlen(n) - 1; i >= 0; i--)
    {
        printf("%c", n[i]);
    }
}

```

```

    }

    return 0;
}

```

ในตัวอย่างเราได้ใช้ for loop ในการวนอ่านค่าจากในตัวแปรอาเรย์ โดยจำนวนรอบในการวนจะใช้ฟังก์ชัน strlen ในการหาความยาวของอาเรย์ ซึ่งคุณจะได้เรียนรู้เรื่องอาเรย์ในบทหลัง โดยในครั้งแรกจะอ่านมาแสดงปกติ และครั้งที่สองจะอ่านจากด้านหลังมา โดยจะได้ผลลัพธ์ดังนี้

Hello World

dlroW olleH

2.3.5.7 คำสั่ง break เป็นคำสั่งที่ใช้สำหรับหยุดการทำงาน loop ในทันที โดยจะไม่ส่งใจ氟ื่นไปของ loop ว่าังเป็นจริงอยู่หรือไม่ มาดูตัวอย่างการใช้งานในการแสดงผลตัวเลข 1 ถึง 10

```

#include <stdio.h>

int main()
{
    int i;

    for (i = 1; i <= 10; i++)
    {
        printf("%d, ", i);

        if (i == 3)
            break;
    }
}

return 0;
}

```

ในการทำงานของโปรแกรม โปรแกรมจะวนเพียง 3 รอบ เพราะว่าเมื่อ i มีค่าเป็น 3 เราได้ใช้คำสั่ง break เพื่อให้สิ้นสุดการทำงานของ for loop ทันที โดยจะไม่ส่งใจ氟ื่นไปของ for loop จะยังคงเป็นจริง และนี้เป็นผลลัพธ์ของโปรแกรมเมื่อรัน

1, 2, 3,

2.3.5.8 คำสั่ง continue เป็นคำสั่งสำหรับการข้ามการทำงานในรอบปัจจุบัน และทำงานในรอบใหม่ทันที มาดูตัวอย่าง

```

#include <stdio.h>

int main()
{
    int i;

    for (i = 1; i <= 10; i++)
    {
        if (i % 2 != 0)
            continue;

        printf("%d, ", i);
    }
}

```

```

    }

    return 0;
}

```

ในตัวอย่างโปรแกรมจะทำงานคำสั่ง continue ถ้าหาก i หาร 2 มีเศษเป็น 0 หรือจำนวนคู่ ดังนั้นโปรแกรมจะได้ดังนี้

2, 4, 6, 8, 10,

ในบทนี้ คุณได้เรียนรู้คำสั่งควบคุมในเบื้องต้นในภาษา C เราได้พูดถึงการควบคุมโปรแกรมด้วยคำสั่งเลือกเงื่อนไขอย่างเช่นคำสั่ง If และ switch และคำสั่งวนซ้ำที่สามารถทำให้โปรแกรมของเรารажาทำงานแบบเดิมซ้ำๆ ได้ภายในเงื่อนไขที่เปลี่ยนแปลงไป และนอกจากนี้คุณยังได้รู้จักการใช้คำสั่ง continue และ break สำหรับควบคุมการทำงานของลูป

2.3.6 พิงก์ชัน

ฟิงก์ชัน คือส่วนของโปรแกรมที่ถูกเขียนขึ้นเพื่อทำหน้าที่เฉพาะของมัน ฟิงก์ชันในภาษา C นั้นจะมีฟิงก์ชันที่มากับภาษา และฟิงก์ชันที่ผู้ใช้สร้างขึ้นเอง ในบทนี้เราจะพูดเกี่ยวกับฟิงก์ชันที่สร้างขึ้นเองเป็นส่วนมาก

2.3.6.1 ฟิงก์ชันในภาษา C มีฟิงก์ชันต่างๆ มากมายที่เราสามารถใช้ได้ เพื่อทำหน้าที่ต่างๆ ที่ต้องการ ในการใช้ฟิงก์ชันโดยปกติต้อง include header file ของฟิงก์ชันนั้นมาด้วย ตารางข้างล่างเป็นตัวอย่างฟิงก์ชันบางส่วนในภาษา C

Function name	Library	Usage
scanf	stdio.h	getting keyboard input
printf	stdio.h	display text to screen
getch	conio.h	get character input
sqrt	math.h	get square root of number
floor	math.h	floor floating point of number to int
strlen	stdio.h	get characters' length

ตัวอย่างการใช้ฟิงก์ชันที่มากับภาษา C

```
sqrt(25); //5
```

```
floor(1.34); //1
```

2.3.6.2 การประกาศฟังก์ชัน เมื่อหาที่เราจะเน้นในบทนี้คือฟังก์ชันที่ผู้ใช้สร้างขึ้นเอง ในภาษา C นั้น คุณสามารถสร้างฟังก์ชันของคุณเอง เพื่อที่จะให้ทำงานตามที่ต้องการได้ แนวคิดของฟังก์ชันนี้คือนำโค้ดกลับมาใช้ใหม่โดยที่ไม่ต้องเขียนโปรแกรมใหม่ทั้งหมด ก่อนที่เราจะใช้ฟังก์ชัน มันจำเป็นต้องถูกประกาศก่อน ซึ่งรูปแบบการประกาศฟังก์ชันในภาษา C:

```
type name (parameter1, parameter2, ...)
```

```
{
```

```
statements
```

```
}
```

type เป็นประเภทของฟังก์ชันที่ต้องการประกาศ ซึ่งขึ้นกับค่าผลลัพธ์ของฟังก์ชันเป็นอะไร คุณสามารถประกาศ function type ได้เหมือนกับ primitive datatype เช่น int, float, char และอื่นๆ สำหรับฟังก์ชันที่ไม่มีการส่งค่ากลับจะมี type เป็น void

name เป็นชื่อของฟังก์ชันที่คุณต้องการสร้าง ในการตั้งชื่อฟังก์ชันมันมีกฎ เช่น เดียวกันเหมือนกับการตั้งชื่อตัวแปร ชื่อของฟังก์ชันเป็นสิ่งที่เราจะใช้ เมื่อต้องการใช้งานฟังก์ชัน

parameters เป็นตัวแปรที่ส่งเข้ามาในฟังก์ชัน พารามิเตอร์เป็นทางเลือกซึ่งสามารถมีหรือไม่ก็ได้

statement เป็นคำสั่งของโปรแกรมเพื่อที่จะให้ฟังก์ชันทำงานและได้ผลลัพธ์ที่ต้องการ

มาดูตัวอย่างของฟังก์ชันหา factorial ของตัวเลขที่เราจะสร้างขึ้นเอง

```
#include <stdio.h>

int myFactorial(int);

int main()
{
    int myNumber = 5;
    printf("%d face is %d", myNumber, myFactorial(myNumber));
    return 0;
}

int myFactorial(int n)
{
    int result = 1, i;
    for (i = 1; i <= n; i++)
    {
        result *= i;
    }
    return result;
}
```

ในตัวอย่างเป็นฟังก์ชันในการหาค่า factorial ของตัวเลข ที่สร้างขึ้นเอง ในการสร้างฟังก์ชันในภาษา C จะต้องกำหนดฟังก์ชันก่อนที่จะเรียกใช้งาน สังเกตุในตัวอย่าง เราได้สร้างฟังก์ชันอยู่ด้านล่าง แต่เราเรียกใช้ฟังก์ชันก่อนที่มีจะถูกสร้างในฟังก์ชัน main ของเรา ดังนั้นมันจึงจำเป็นต้องถูกประกาศไว้ก่อนด้านบนของโปรแกรม และนี้เป็นผลลัพธ์ของโปรแกรม

5 face is 120

2.3.6.3 ฟังก์ชันพารามิเตอร์เป็นพารามิเตอร์ที่ส่งเข้ามาในฟังก์ชัน โดยจะเป็นลำดับของตัวแปรที่ส่งเข้ามา พารามิเตอร์จะมีความต้องพึ่งพา กับอาร์กิวเม้นต์ของฟังก์ชัน จากตัวอย่างก่อนหน้า ในฟังก์ชันหา factorial นั้นมีพารามิเตอร์คือ n มาดูตัวอย่างของฟังก์ชันพารามิเตอร์

```
#include <stdio.h>

float area(float, float);

int main()
{
    float width = 8;
    float height = 5.4;
    printf("Area is %.2f", area(width, height));
    return 0;
}

float area(float w, float h)
{
    return w * h;
}
```

ในตัวอย่าง เราได้สร้างฟังก์ชันในการหาพื้นที่ของกล่องสี่เหลี่ยม ในการที่จะหาพื้นที่เราต้องการความกว้างและความยาว สังเกตว่าพารามิเตอร์ของฟังก์ชันนี้จะมีสองอัน คือ w และ h ซึ่งมีประเภทเป็น float และฟังก์ชันนี้ได้ return ค่าที่เป็น float กลับไปเข่นกัน เมื่อรันโปรแกรมจะได้ผลลัพธ์ดังนี้

Area is 43.20

นอกจากนี้ยังสามารถที่จะไม่มีพารามิเตอร์ได้

2.3.6.4 ฟังก์ชันอาร์กิวเม้นต์ เป็นค่าของตัวแปรที่จะส่งเข้าไปในฟังก์ชัน ซึ่งจะต้องสอดคล้องกับฟังก์ชันพารามิเตอร์ ในตัวอย่างของคำสั่ง area(width, height) สังเกตว่าฟังก์ชันอาร์กิวเม้นต์จะมีสองตัว และประเภทของฟังก์ชันจะตรงกับลำดับของฟังก์ชันพารามิเตอร์ที่เราได้สร้างไว้

2.3.6.5 คำ return ของฟังก์ชันในการเรียนฟังก์ชันนั้น ส่วนมากจำเป็นต้องมีการส่งค่ากลับ โดยการใช้คำสั่ง return โดยในการประกาศฟังก์ชัน ประเภทของฟังก์ชันบ่งบอกถึงประเภทค่าที่ต้องการส่งกลับ จากในตัวอย่าง

```
float area(float, float);
```

สังเกตว่า function type จะเป็น float นั่นหมายความว่าฟังก์ชันนี้จะมีค่าที่ส่งกลับเป็นเลขจำนวนจริง

นอกจากนี้ ในภาษา C เราสามารถสร้างฟังก์ชันที่ไม่มีคำ return ได้ โดยให้มี function type เป็น void มาดูตัวอย่าง

```
#include <stdio.h>

void introduce(char s[])
{
    printf("My name is %s\n", s);
}

void sayHello(char s[])
{
    printf("Hello %s\n", s);
}

int main()
{
    char name[] = "Mateo";
    introduce(name);
    sayHello("Tommy");
    return 0;
}
```

ผลลัพธ์เมื่อรันโปรแกรม

```
My name is Mateo
Hello Tommy
```

2.4 อุปกรณ์ Arduino

บอร์ด Arduino เป็นไมโครคอนโทรลเลอร์ ที่สามารถอ่านอินพุตจากตัวตรวจจับแสง, ใช้nickดูบันปุ่ม หรือส่งข้อความไปยัง Twitter และเปลี่ยนเป็นເອົາຕຸພູດໃຊ້ງານມອເຕອຣ໌, เปิดໄຟ LED ທີ່ຮູ້ແຜຣ່ຂໍອມລູໄປຍ່ງຮະບບອິນເທິວົນເນື້ອຕີ້ກັດວ່າ ຈຶ່ງຜູ້ໃຊ້ງານສາມາຮັດຄວບຄຸມບອດວ່າ ຕ້ອງທຳວ່າໄຮ ໂດຍສ່າງຊຸດຄໍາສໍ້ໄປຢ່າງໄມ້ໂຄຣຄອນໂກຣລເລອ່ຽນບອດ ໃນການທຳເຊັ່ນນັ້ນຄຸນຕ້ອງໃຫ້ກາຫາ Arduino ຈຶ່ງມີຄໍາສໍ້ເພີ່ມເຂັ້ມາເພື່ອເຂີຍໃນຮູບ ແບບກາຫາ C++ ແລະໃໝ່ໂຟຣ໌Arduino IDE ເປັນຫັກໃນການປະມວລຜລ

ໃນຊ່ວງໜ່າຍປີທີ່ຜ່ານມາ Arduino ເປັນສ່ວນຫັກຂອງໂຄຣກາມກາມຍ້າງ ຕັ້ງແຕ່ວັດຖຸປະສົງປະຈຳວັນໄປຈົນຄົງເຄື່ອງມືອົວທີ່ຢາສຕົກ ຊັບຊ້ອນ ຊຸມໜຸນອອນໄລ່ນໍ ຂອງ Maker ທ່ານໄລ ຈຶ່ງມີ ນັກເຂີຍ/ນັກສຶກສາ, ຜູ້ຂອບທໍາງານອົດິເຣີກ, ສີລົບິນ, ນັກເຂີຍໂປຣແກຣມແລະຜູ້ເຊີ່ວໜາຍ ໄດ້ຮັມຕັກນັ້ນ ໃຊ້ງານສໍາຫຼັບ ແພລຕົວໝ່າງແບບເປີດນີ້ ການມີສ່ວນຮ່ວມຂອງພວກເຂົາໄດ້ເພີ່ມຄວາມຮູ້ທ່ານເຂົ້າສົ່ງໄດ້ຢ່າງເຫຼືອເຂົ້າສົ່ງສາມາດເປັນປະໂຍ້ນຍ່າງນາກສໍາຫຼັບ ມີໂທ່າງແລະຜູ້ເຊີ່ວໜາຍ

Arduino ມີຈຸດກຳນົດເຮີມຕັ້ນເຂົ້າທີ່ສາບັນກາຮອກແບບປະລິສັມພັນຮ໌ Ivrea ປະເທດອິຕາລີ ຈຶ່ງເປັນເຄື່ອງມືອີ່ນໆທີ່່ງ່າຍສໍາຫຼັບກາຮສ້າງຕັ້ນແບບທີ່ ຮວດເວົ່າມີເປົ້າໄປທີ່ນັກເຂີຍທີ່ມີເພື່ອກັນດ້ານອີເລີກທຣອນິກສີ ແລະກາຮເຂີຍໂປຣແກຣມ ແຕ່ກົມືຜູ້ໃຊ້ໜ່າຍຄົນພາຍາມນຳ Arduino ໄປໃຊ້ໃນຮະບບງານຈິງ ຖ້າ ທັ້ງນັ້ນຂຶ້ນອູ້ກັບປັ້ງຫາແລະຄວາມຍາກຈ່າຍຂອງງານນັ້ນ ຖ້າ Arduino ເປັນມີກາຮເປົ້າຍແປ່ງແລ້ວເປົ້າໃຫ້ກັບຄວາມຕ້ອງກາຮແລະຄວາມທ້າທາຍໃໝ່ ຈາກບອດ 8 ບົດແບບຈ່າຍ ຈຶ່ງກັບຜົດກິລິກັນທີ່ສໍາຫຼັບແອປພິເຄີເຂົ້ນສໍາຫຼັບ IoT, ອຸປະກິນສ່າມໄສ, ເຄື່ອງທີມພໍ 3 ມິຕີ ແລະສກາພແວດລ້ອມແບບຝຶ່ງຕ້ວ

2.4.1 ภาษาซีของ Arduino

ภาษาซีของ Arduino จะจัดแบ่งรูปแบบโครงสร้างของการเขียนโปรแกรมออกเป็นส่วนย่อยๆ หลายส่วน โดยเรียกแต่ละส่วนว่า พังก์ชั่น และ เมื่อนำพังก์ชั่น มารวมเข้าด้วยกัน ก็จะเรียกว่า โปรแกรม โดยโครงสร้างการเขียนโปรแกรมของ Arduino นั้น ทุกๆ โปรแกรมจะต้องประกอบไปด้วยพังก์ชั่นจำนวน เท่าเดียวกัน แต่ถ้าต้องการให้สุดตัวมีพังก์ชั่น จำนวน 2 พังก์ชั่น คือ `setup()` และ `loop()` ดังต่อไปนี้

```
#include <Servo.h> // ส่งผ่านไฟล์ ชื่อ Servo.h เข้ามายังไฟล์โปรแกรม

int Servo1 = 9; // กำหนดให้ Servo1 แทน Pin Digital-9

Servo myservo; // สร้าง object ชื่อ myservo เพื่อควบคุม Servo

void setup() {

    myservo.attach(Servo1); // กำหนดให้ใช้ขา Digital-9 สร้างสัญญาณควบคุม
}

void loop() {

    myservo.write(180); // กำหนดค่าตำแหน่งให้กับ Servo = 180 องศา
}
```

จะเห็นได้ว่าโครงสร้างพื้นฐานของภาษาซีที่ใช้กับ Arduino นั้น จะประกอบไปด้วย 3 ส่วนใหญ่ๆ ด้วยกัน คือ

- header ในส่วนนี้จะมีหรือไม่มีก็ได้ ถ้ามีต้องกำหนดให้ในส่วนเริ่มนั้นของโปรแกรม ซึ่งส่วนของ Header ได้แก่ ส่วนที่เป็น Compiler Directive ต่างๆ รวมไปถึงส่วนของการประกาศตัวแปร และ ค่าคงที่ต่างๆ ที่จะใช้ในโปรแกรม
- `setup()` ในส่วนนี้เป็นพังก์ชั่นบังคับที่ต้องกำหนดให้มีในทุกๆ โปรแกรม ถึงแม้ว่าในบางโปรแกรมจะไม่ต้องการใช้งานก็ยัง จำเป็นต้องประกาศไว้ด้วยเสมอ เพียงแต่ไม่ต้องเขียนคำสั่งใดๆ ไว้ในระหว่าง วงล้อบีก้า { } ที่ใช้เป็นตัวกำหนดขอบเขตของพังก์ชั่น โดยพังก์ชั่นนี้จะใช้สำหรับระบุค่าสั่งใน ส่วนที่ต้องการให้โปรแกรม ทำงานพึ่งรับเดียวตอนเริ่มนั้นทำงานของโปรแกรมครั้งแรกเท่านั้น ซึ่ง ได้แก่คำสั่งเกี่ยวกับการ `Setup` ค่าการทำงานต่างๆ เช่น การกำหนดหน้าที่การใช้งานของ `PinMode` และการกำหนดค่า `Baudrate` สำหรับใช้งานพอร์ตสื่อสารอนุกรม เป็นต้น
- `loop()` เป็นส่วนพังก์ชั่นบังคับที่ต้องกำหนดให้มีในทุกๆ โปรแกรม เช่นเดียวกับพังก์ชั่น `setup()` โดยพังก์ชั่น `loop()` นี้จะใช้บรรจุ คำสั่งที่ต้องการให้โปรแกรมทำงานเป็นวงรอบซ้ำๆ กันไปไม่รู้จบ ซึ่ง ถ้าเปรียบเทียบกับรูปแบบของ ANSI-C ส่วนนี้ก็คือ พังก์ชั่น `main()` นั่นเอง

2.4.2 ตัวแปรของ Arduino

2.4.2.1 ตัวแปร หมายถึง กลุ่มของ ตัวอักษร ตัวเลข และ เครื่องหมายใดๆ ที่รวมกันเป็นชื่อ เพื่อใช้กำหนด เป็นตัวแทนของค่าข้อมูลที่ เราต้องการจะอ้างถึงในโปรแกรม ทั้งนี้ก็เนื่องจากว่าในการทำงานของโปรแกรม จริงๆ นั้นจะใช้ค่าตัวเลขที่ผู้ใช้กำหนดให้ มาทำการประมวลผล ซึ่ง ในการเขียนโปรแกรมถ้าเราต้องเขียน โปรแกรมโดยกำหนดเป็นค่าตัวเลขให้กับโปรแกรมตรงๆ เลย ก็จะทำให้โปรแกรมที่เราเขียนขึ้นเต็มไปด้วยค่า ตัวเลขต่างๆ มากมาย ซึ่งยากต่อการอ่าน ยากต่อการทำความเข้าใจ และยากต่อการตรวจสอบความถูกต้อง และอาจ ทำให้เกิดความผิดพลาดได้เจ้ายิ่ง ดังนั้นทุกภาษา จึงยอมให้มีการกำหนดชื่อ ขึ้นมาใช้งานแทน ค่าตัวเลข เพื่อให้เขียนโปรแกรมได้สะดวกและง่าย ต่อการอ่าน ทำความเข้าใจ ได้มากยิ่งขึ้น

ซึ่งลักษณะของข้อมูล อาจมีทั้งแบบที่เป็นค่าซึ่งสามารถเปลี่ยนแปลงได้ (variable) หรือ อาจเป็น แบบที่มีค่าคงที่ไม่สามารถเปลี่ยนแปลงได้ (constant) ในการประกาศใช้งานตัวแปร จำเป็นต้องประกาศ ชนิดของตัวแปร หรือบางครั้งอาจมีการกำหนดค่าเริ่มต้นให้กับตัวแปรด้วยก็ได้

2.4.2.2 ชนิดและประเภทของตัวแปร ถ้าหากว่าเราจะเปรียบเทียบว่า ตัวแปร คือ ภาษาที่สำหรับบรรจุสิ่งของ และ ข้อมูล คือ สิ่งของที่เรา ต้องการจะเก็บ จะเห็นได้ว่า สิ่งของต่างๆรอบๆตัวเรานั้น จะมีคุณสมบัติที่แตกต่างกันไป ดังนั้นในการเลือก ภาษาระบบสำหรับใช้บรรจุสิ่งของ เรา ก็จำเป็นต้องเลือกชนิดของภาษาระบบให้มีความเหมาะสมที่จะใช้เก็บสิ่งของ ด้วย ซึ่งสิ่งแรกที่ต้องพิจารณาคือ เราจะต้องรู้ว่าคุณสมบัติของสิ่งของที่ต้องการจะจัดเก็บ และ จุดประสงค์ การใช้งาน ก่อน จากนั้นจึงจัดทำภาษาที่มี ขนาด และ รูปทรงของภาษาระบบ เหมาะสมที่จะใช้เก็บสิ่งของ เพื่อให้สามารถจัดเก็บ และ นำสิ่งของออกมาใช้งานได้อย่างเข้าใจมากที่สุด

ในภาษาซีนั้น มีการกำหนด และ จำแนก ชนิดของตัวแปร ไว้เป็น 5 ชนิดด้วยกัน โดยแต่ละชนิดจะมี คุณสมบัติการใช้งานที่ต่างกัน เพื่อใช้ในการเก็บข้อมูลที่มีรูปแบบแตกต่างกัน คือ

- char ใช้เก็บข้อมูลที่เป็นตัวอักษร (character) ใช้เก็บข้อมูลที่เป็นเลขจำนวนเต็มได้ 256 ค่า
- int ใช้เก็บข้อมูลที่เป็นเลขจำนวนเต็ม (integer) ใช้เก็บข้อมูลที่เป็นเลขจำนวนเต็มได้ 65536 ค่า
- float ใช้เก็บข้อมูลที่เป็นเลขทศนิยมแบบ Single Precision
- double ใช้เก็บข้อมูลที่เป็นเลขทศนิยมแบบ Double Precision ซึ่ง สามารถเก็บค่าตัวเลขทศนิยมที่มีความละเอียดและถูกต้องของหน่วยมากกว่าแบบ float ถึง 2 เท่า
- void ใช้เก็บตัวแปรที่ไม่มีค่า

2.4.2.3 คุณสมบัติเฉพาะของตัวแปร สำหรับตัวแปรชนิดที่ใช้เก็บค่าเลขจำนวนเต็ม (char และ int) นั้น ในภาษาซี ไม่ได้มีการจำแนกชนิด ของตัวแปรเพื่อใช้เก็บค่าตัวเลขที่เป็น ค่าวาก หรือ ค่าลบ เป็นการเฉพาะ แต่ภาษาซี จะใช้วิธีการเพิ่ม คำสั่ง สำหรับกำหนดคุณสมบัติเฉพาะให้กับตัวแปรไว้อีก 4 คำสั่ง สำหรับใช้กำหนดคุณสมบัติของตัวแปรแบบนี้ ให้มีคุณสมบัติที่เฉพาะเจาะจงลงไปอีก เป็นต้นว่า จะใช้ตัวแปรในการเก็บค่าตัวเลขที่เป็นค่าวากอย่างเดียว หรือต้องการเก็บค่าตัวเลขแบบบิตเดียว หรือต้องการเก็บค่าตัวเลขแบบบิตเดียว เพื่อให้ผู้ใช้สามารถปรับแก้ คุณสมบัติในการใช้งานของ ตัวแปรให้มีคุณสมบัติใกล้เคียงกับความต้องการใช้งานมากขึ้นไปอีก และเพื่อจำกัดขอบเขตการใช้งานของ ตัวแปรให้ตรง กับจุดประสงค์มากยิ่งขึ้น และยังเป็นการช่วยให้ประยุกต์จำนวนของหน่วยความจำที่ใช้สร้าง ตัวแปรด้วยและ ทำให้โปรแกรมทำงานได้เร็วขึ้นกว่าเดิมอีกด้วย โดยในภาษาซีจะมีคำสั่ง ที่ใช้สำหรับบุ คุณสมบัติเฉพาะของตัวแปรที่ใช้เก็บค่าเลขจำนวนเต็ม 4 คำสั่ง คือ

- unSigned ใช้ระบุให้เก็บค่าเลขจำนวนเต็มในตัวแปรเฉพาะค่าที่เป็นบวกเท่านั้น
- signed ใช้ระบุให้เก็บค่าเลขจำนวนเต็มในตัวแปรทั้งค่า บวก และ ลบ
- short ใช้ระบุให้เก็บค่าเลขจำนวนเต็มในตัวแปรที่มีค่าน้อยกว่า int
- long ใช้ระบุให้เก็บค่าเลขจำนวนเต็มในตัวแปรที่มีค่านากกว่า int เป็น 2 เท่า

ชนิดข้อมูล	การเก็บข้อมูล	ขนาด
boolean	จริง (True) หรือ เท็จ (False)	1 บิต
char	ตัวเลข หรือตัวอักษร	1 ไบต์ ใส่ค่าได้ตั้งแต่ -128 ถึง 127
unsigned char	ตัวเลข หรือตัวอักษร	1 ไบต์ ใส่ค่าได้ตั้งแต่ 0 ถึง 255
byte	ไบต์	1 ไบต์ ใส่ค่าได้ตั้งแต่ 0 ถึง 255
int	ตัวเลขจำนวนเต็ม	2 ไบต์ ใส่ค่าได้ตั้งแต่ -32,768 ถึง 32,767

unsigned int	ตัวเลขจำนวนเต็ม	2 ไบต์ ใส่ค่าได้ตั้งแต่ 0 ถึง 65,535 (2^{16}) - 1
long	ตัวเลขจำนวนเต็มที่มีความยาว	4 ไบต์ ใส่ค่าได้ตั้งแต่ -2,147,483,648 ถึง 2,147,483,647
unsigned long	ตัวเลขจำนวนเต็มที่มีความยาว	4 ไบต์ ใส่ค่าได้ตั้งแต่ 0 ถึง 4,294,967,295 (2^{32} - 1)
float	ตัวเลขทศนิยมใช้ในการคำนวณ	4 ไบต์ ใส่ค่าได้ตั้งแต่ 3.4028235E+38 ถึง -3.4028235E+38 มีทศนิยมได้ 6 ถึง 7 ตำแหน่ง
double (เฉพาะบอร์ด Arduino Due)	ตัวเลขทศนิยมที่มีความยาวและต้องการความแม่นยำ	8 ไบต์ ใช้ในการคำนวณที่ต้องการประสิทธิภาพสูง
String	ข้อความ	ไม่ระบุ

2.4.2.4 การประกาศตัวแปร เช็ตค่าตัวแปร การประกาศตัวแปรจะเหมือนกับภาษา C โดยปกติ คือ

TYPE KEY;

โดย TYPE เป็นชนิดของข้อมูล ส่วน KEY เป็นชื่อตัวแปร การประกาศตัวแปรข้างต้นคือการประกาศตัวแปรแบบไม่กำหนดค่า ดังนั้นค่าปกติที่อ่านจากตัวแปรจะเป็น 0

TYPE KEY = VAL;

จากตัวอย่างด้านบน เป็นลักษณะของการประกาศตัวแปรแบบกำหนดค่า เมื่ออ่านค่าของตัวแปรออกมานะ จะได้เป็นค่าที่กำหนดไว้ตอนประกาศ

ตัวอย่างการประกาศตัวแปร

```
int i;
int a = 10, b = 20;
```

จากตัวอย่าง จะเห็นว่าความสามารถกำหนดชนิดของข้อมูลให้ต้องเป็นทางเดียวทั้งๆที่ตัวแปรก็ได้ โดยใช้เครื่องหมาย , คันทรี

```
int i;
int a = 10, b = 20;
i = a + b;
```

จากตัวอย่างด้านบน เราสามารถกำหนดค่าให้กับตัวแปรเมื่อแรกๆได้ โดยใช้เครื่องหมายเท่ากับ = เป็นตัวเชื่อม ชื่อตัวแปรจะอยู่ทางซ้าย และจะกำหนดค่าเป็นอะไร ให้อยู่ทางขวา ค่าที่อยู่ทางขวาจะถูกนำมายังในค่าที่อยู่ทางซ้ายเสมอ

```
int i = 10, a;
a = i;
```

จากตัวอย่างด้านบน จะเห็นว่า a ไม่ได้กำหนดค่าไว้ตอนประกาศ ทำให้ค่าที่อ่านได้จาก a คือ 0 แต่บรรทัดถัดมา มีการกำหนดค่าให้ a เท่ากับ i ซึ่งตอนประกาศ i ได้ประกาศไว้ว่าค่าเท่ากับ 10 เมื่อนำมาใส่ a ค่าที่อ่านได้จาก a จึงเป็น 10 ด้วยเช่นกัน

```
boolean is = false;
is = !is;
```

จากตัวอย่างด้านบน มีการประกาศตัวแปร boolean ซึ่งเป็นตัวแปรทางลوجิก มีค่าเป็น True (1) หรือ False (0) ได้เท่านั้น ในบรรทัดแรกได้ประกาศว่าตัวแปร is เป็นตัวแปรชนิด boolean และมีค่าเป็น false หรือล็อกิค 0 บรรทัดต่อมา ได้มีการกำหนดให้ is เท่ากับ !is การที่เครื่องหมายนี้เสริปอยู่หน้าตัวแปร หมายถึงการกลับเป็นค่าตรงข้าม จากบรรทัดแรก ตัวแปร is มีค่าเป็น false เมื่อเจอ !is ค่าจึงถูกกลับเป็น true และถูกนำไปใช้ในตัวแปร is ทำให้สุดท้ายแล้วตัวแปร is มีค่าเป็น true

```
String text = "Myarduino";
```

จากตัวอย่างด้านบน ได้มีการประกาศตัวแปรชื่อ text เป็นชนิด String เมื่ออ่านค่าที่ได้จาก text จึงได้ค่าอ กมาเป็น "Myarduino" เลย

* การกำหนดค่าแบบข้อความให้กับตัวแปร จะต้องอยู่ภายใต้เครื่องหมาย "" เท่านั้น มิฉนั้นโปรแกรมจะแสดงข้อความผิดพลาดอ กมา

2.4.2.5 สรุปชนิดของตัวแปรใน Arduino ที่ใช้บ่อยๆ

- boolean ใช้เก็บค่าข้อมูล เพียง 2 จำนวน คือ TRUE (จริง) และ FALSE (เท็จ)
- char ใช้เก็บค่าข้อมูลขนาด 8 บิต ใช้สำหรับเก็บค่ารหัสของตัวอักษร ซึ่งสามารถกำหนดเป็นค่า หรือ เปลี่ยนตัวอักษรไว้ภายใต้เครื่องหมาย พื้นเดียวกันได้ เช่น 'A' หรือ 0x41 หรือ 65
- byte ใช้เก็บค่าข้อมูลขนาด 8 บิตที่เป็นค่าจำนวนเต็มแบบไม่คิดเครื่องหมาย เหมือนกันกับชนิด signed char ในภาษาซี ซึ่งสามารถเก็บค่าข้อมูลได้ 256 ค่า คือ 0-255
- int หรือ Integer ใช้เก็บค่าข้อมูลขนาด 16บิต ที่เป็นค่าจำนวนเต็ม แบบคิดเครื่องหมาย โดยสามารถใช้เก็บข้อมูลได้ 65536 ค่า คือ -32768 ถึง +32767
- unsigned int ใช้เก็บค่าข้อมูลขนาด 16บิต ที่เป็นค่าจำนวนเต็ม แบบไม่คิดเครื่องหมาย โดยสามารถใช้เก็บข้อมูลได้ 65536 ค่า คือ 0-65535
- long ใช้เก็บค่าข้อมูลขนาด 32บิต ที่เป็นค่าเลขจำนวนเต็มแบบคิดเครื่องหมาย โดยสามารถใช้เก็บข้อมูลได้ 4294967296 ค่า คือ -2,147,483,648 ถึง 2,147,483,647
- unsigned long ใช้เก็บค่าข้อมูลขนาด 32 บิต ที่เป็นค่าเลขจำนวนเต็มแบบไม่คิดเครื่องหมาย โดยสามารถใช้เก็บข้อมูลได้ 4294967296 ค่า คือ 0 ถึง 4,294,967,295
- float ใช้เก็บค่าข้อมูลที่เป็นเลขทศนิยมแบบคิดเครื่องหมายขนาด 32 บิต โดยสามารถเก็บค่าได้ ระหว่าง 3.4E-38 ถึง 3.4E+38 (-3.4028235E+38 ถึง 3.4028235E+38)
- double ใช้เก็บค่า เลขทศนิยมเช่นเดียวกันกับ float แต่มีค่าความละเอียดกว่า float ถึง 2 เท่า สามารถเก็บค่าได้มากถึง 1.7E+308
- void เป็นตัวแปรแบบที่ไม่มีการเก็บค่าใดๆ คือ ไม่มีค่านั้นเอง
- arrays เป็นตัวแปรที่ใช้เก็บข้อมูลหลายค่าไว้ในตัวแปรตัวเพียงชื่อเดียว แต่มีตัวเลขสำหรับซึ่งตำแหน่งการเก็บข้อมูลต่างกัน โดยตัวเลขที่ใช้พิเศษนั้นที่เป็นตัวซึ่งตำแหน่งของข้อมูล เรียกว่า Index Number โดยค่าลำดับของข้อมูลในตัวแปร array ตำแหน่งแรกจะมีค่าเป็นศูนย์เสมอ
- string เป็นตัวแปรใช้เก็บข้อความ หรือ ตัวอักษรหลายตัว ซึ่ง string ก็คือ array ของตัวแปร แบบ char นั่นเอง
- pointer เป็นตัวแปรที่ไม่ได้ใช้เก็บข้อมูล แต่ใช้เก็บค่าตำแหน่งของข้อมูล เรียกว่า Index Number โดยค่าลำดับของข้อมูลในตัวแปร array ตำแหน่งแรกจะมีค่าเป็นตัวแปรแบบนี้จะใช้พิเศษนั้นที่เป็นตัวซึ่งไปยังตำแหน่งของข้อมูล ของตัวแปรอื่นๆอีกที่หนึ่ง

2.4.3 คำสั่งควบคุมของ Arduino

2.4.3.1 คำสั่ง if เป็นคำสั่งใช้สำหรับตรวจสอบเงื่อนไข เพื่อสั่งให้โปรแกรมเลือกทำงานในปีกการทำงานต่างๆ ตามผลลัพธ์ที่ได้จากการตรวจสอบเงื่อนไขของคำสั่ง โดยมีรูปแบบดังนี้

รูปแบบที่ 1 เป็นแบบ If

```
if (เงื่อนไข) {
    คำสั่งที่ต้องการให้ทำเมื่อเงื่อนไขเป็นจริง
}
```

ตัวอย่างคำสั่ง if

```
int i = 1;
if (i == 1) {
    ถ้า i เท่ากับ 1 จะทำใน ปีก้านี้
}
```

รูปแบบที่ 2 เป็นแบบ If...else

```
if (เงื่อนไข) {
    คำสั่งที่ต้องการให้ทำเมื่อเงื่อนไขเป็นจริง
}
else {
    คำสั่งที่ต้องการให้ทำเมื่อเงื่อนไขเป็นจริงเท็จ
}
```

ตัวอย่างคำสั่ง If...else

```
int x = 1 ;
if (x == 1) {
    ถ้า x เท่ากับ 1 จะทำในปีก้านี้ ถ้าไม่เท่ากับ 1 จะข้ามไปทำในปีก้า Else
}
else {
    ถ้า x ไม่เท่ากับ 1 จะทำในปีก้านี้
}
```

คำสั่งที่ใช้ตรวจสอบเงื่อนไข ได้แก่

- == หมายความว่า เท่ากับ
- != หมายความว่า ไม่เท่ากับ
- < หมายความว่า น้อยกว่า
- > หมายความว่า มากกว่า
- <= หมายความว่า น้อยกว่าหรือเท่ากับ

- $>=$ หมายความว่า มากกว่าหรือเท่ากับ
- $\&\&$ หมายความว่า และ
- $\|$ หมายความว่า หรือ

2.4.4 Arduino Sensor

เนื่องจากลักษณะของโอเพนซอร์ส Arduino จึงกลายเป็นปรากฏการณ์ระดับโลก พัฒนาเป็นแพลตฟอร์มต้นแบบสำหรับมือสมัครเล่น ศิลปินนักออกแบบที่สำคัญไปกว่านั้นสำหรับนักเรียนที่ยังไม่ต่อโลกของโครงการอิเล็กทรอนิกส์Arduino มาพร้อมกับไมโครคอนโทรลเลอร์และซอฟต์แวร์ IDE เพื่อช่วยให้ลดเวลาในการติดตั้งบอร์ดฮาร์ดแวร์ ผู้เรียนสามารถเรียนรู้การใช้งานของ Arduino ในหมู่นักเรียนอื่นๆได้โดยทันที ทำให้ Arduino จึงได้เปิดตัวมีเชื่อมโยงกับ Arduino หลากหลายในตลาด เช่นเซอร์เพลน้ำซึ่งสามารถติดต่อกับบอร์ดและออกแบบแอปพลิเคชันใหม่ ๆ

2.4.4.1 หลักการทำงาน ไมโครคอนโทรลเลอร์ที่มา ก่อน Arduino ไม่มีซอฟต์แวร์ IDE สำหรับอัปโหลดโค้ดลงในบอร์ดแวร์ หนึ่งต้องใช้อุปกรณ์ฮาร์ดแวร์แยกต่างหากเพื่ออัปโหลดรหัสลงในบอร์ดแวร์ เนื่องจากคุณสมบัติความเรียบง่ายจึงง่ายต่อการเขียนต่อเชื่อมต่อเซ็นเซอร์กับ Arduino เนื่องจากไมโครคอนโทรลเลอร์มีซอฟต์แวร์ IDE สำหรับการเขียนโปรแกรมอย่างอิสระสามารถเขียนที่จำเป็นในการเขียนต่อเซ็นเซอร์เหล่านี้กับ Arduino คือ Breadboard และสายเชื่อมต่อโดยสามารถเขียนใน Arduino IDE และอัปโหลดได้ จำเป็นต้องมีแหล่งจ่ายไฟกราว์ดบอร์ดและสายเชื่อมต่อสำหรับการเขียนต่อ

2.4.4.2 การใช้งาน Arduino Sensor มีหลักการที่ออกแบบโดยใช้เซ็นเซอร์ Arduino สำหรับการใช้งานต่างๆ กล่าวกันว่า Arduino ถูกนำมาใช้เพื่อสร้างความคิดในฝันให้เป็นจริง โดยอุดมคติในการทำงาน ไม่ต้องมีอุปกรณ์ใดๆ ให้สนใจในการทำงาน เช่นเซอร์หลักเลี้ยงสิ่งกีดขวาง IR อินฟราเรดตรวจจับวัตถุที่อยู่ก่อนหน้าและสร้างสัญญาณดิจิตอล ให้ในทุนยนต์ Soil hygrometer คือเซ็นเซอร์วัดความชื้นในดิน สร้างสัญญาณดิจิตอลเมื่อความชื้นในดินเพิ่มขึ้นสูงกว่าค่าเกณฑ์บางอย่าง โรงงานน้ำอัดโนมัติได้รับการออกแบบโดยใช้เซ็นเซอร์นี้กับ Arduino เช่นเซอร์กล้องจูลทรรศน์ใช้ในการตรวจสอบความสะอาด เช่นเซอร์ความชื้นในดินเพิ่มขึ้นของเสียงที่ตราชพบเพิ่มขึ้นเกินค่าเกณฑ์บางค่า Digital Barometric Pressure Sensor ใช้ในการวัดความดันสัมบูรณ์ของสิ่งแวดล้อม สามารถวัดความสูงของทุ่นยนต์หรือกระสุนปืนได้โดยใช้เซ็นเซอร์นี้ สำหรับการตรวจจับแสงจะใช้โมดูลเซ็นเซอร์ไฟฟ้าเรซิสเตอร์ ระบบไฟรักษาความปลอดภัยกลางคืนใช้เซ็นเซอร์นี้กับ Arduino เช่นเซอร์อุณหภูมิใช้เพื่อตรวจจับอุณหภูมิโดยรอบในการตรวจจับก๊าซพิษ เช่น LPG, i-Butane, Propane, Alcohol ฯลฯ ... ใช้เซ็นเซอร์ก๊าซ MQ-2 เช่นเซอร์วัดปริมาณน้ำฝนใช้สำหรับการตรวจสอบสภาพอากาศ ในการตรวจจับเปลวไฟและใช้เซ็นเซอร์เปลวไฟธรรมชาติ เช่นเซอร์ PIR ใช้เพื่อตรวจจับการเคลื่อนไหว จากระบบเซ็นเซอร์และสั่นสะเทือน เช่นเซอร์หน้าจอสัมผัสใช้สำหรับออกแบบวงจรหรือแสดงโดยใช้ Arduino

2.4.4.3 Arduino Sensor ปัจจุบัน เช่นเซอร์ Arduino มีหลายประเภท ดังนี้

- โมดูลอัลตราโซนิก HC-SR04
- IR อินฟราเรดเซ็นเซอร์หลักเลี้ยงอุปสรรค
- ติน ไโกรมิเตอร์ โมดูลการตรวจจับ
- เช็นเซอร์ความชื้นในดิน
- เช็นเซอร์ไมโครโฟน
- ดิจิตอลเซ็นเซอร์ความดันบรรยากาศ
- เช็นเซอร์ไฟฟ้าเรซิสเตอร์
- เช็นเซอร์ความร้อนดิจิตอล - เช็นเซอร์อุณหภูมิ
- โมดูลเข้ารหัสแบบหมุน
- MQ-2 เช็นเซอร์แก๊ส

- SW-420 เซ็นเซอร์ตรวจจับความเคลื่อนไหว
- เซ็นเซอร์ตรวจจับความชื้นและฝน
- ไมค์ล Passive Buzzer
- ไมค์ลเซ็นเซอร์ความเร็ว
- เซ็นเซอร์ตรวจจับเปลวไฟอินฟราเรด IR
- ไมค์ลรีเลย์ช่องสัญญาณ 5V 2 ช่อง
- ไมค์ลพาวเวอร์ชัพพลาຍ Breadboard 3.3V
- เซ็นเซอร์อินฟราเรด Pyroelectric HC-SR501
- ไมค์ล Accelerometer
- DHT11 เซ็นเซอร์อุณหภูมิและความชื้น
- เครื่องส่ง / ตัวรับ RF 433MHz

เซ็นเซอร์ Arduino เหล่านี้ทำให้การดำเนินโครงการอิเล็กทรอนิกส์เป็นไปได้มากมาย ตัวอย่างบางส่วนของโครงการดังกล่าว ได้แก่ Suntracker โดยใช้ LDR กับ Arduino, สัญญาณเตือนน้ำฝน Arduino, ทุบยันต์ควบคุมด้วยท่าทางที่ใช้ Accelerometer พร้อม Arduino, ผู้ติดตามสายที่ใช้เซ็นเซอร์ IR, สัญญาณเตือนการตรวจจับการเคลื่อนไหวตามเซ็นเซอร์ IR, สัญญาณเตือนประตูโดยใช้เซ็นเซอร์อัลตราโซนิก, การวัดระยะทางโดยใช้เซ็นเซอร์อัลตราโซนิก, แต่งตาบอดอัจฉริยะที่ใช้ Arduino, เซ็นเซอร์ PIR สำหรับควบคุมเครื่องใช้ในบ้านโดยใช้ Arduino ฯลฯ Arduino เป็นตัวเลือกแรกสำหรับนักเรียนและผู้ที่ไม่คุ้นเคยกับอุปกรณ์อิเล็กทรอนิกส์ในการออกแบบโครงการใหม่ เซ็นเซอร์ Arduino สามารถใช้กับตัวอื่น ๆ ได้ไม่ยากบนโลกArduino เช่นกัน. Arduino IDE ประกอบด้วยไลบรารีต่างๆที่มีประโยชน์ในการเขียนโปรแกรมต่อเซ็นเซอร์ประเภทต่างๆ ข้อยกเว้นเพียงประการเดียวคือเซ็นเซอร์ที่ต้องการความเร็วในการประมวลผลมากกว่าที่ Arduino สามารถให้ได้ คุณสามารถเขียนต่อเซ็นเซอร์ได้ก่อตัวต่อครั้งบนบอร์ด Arduino ตัวเดียว?

แหล่งอ้างอิงข้อมูลจาก [บทเรียนภาษา C, สอนภาษา C เนื้องต้น - MarcusCode](#)

[บหความ - ขาย Arduino อุปกรณ์ Arduino คุณภาพดี ราคาถูก ส่งไว ส่งฟรี \(cybertice.com\)](#)

[ความรู้เกี่ยวกับคนพิการ – งานบริการนักศึกษาพิการ \(stou.ac.th\)](#)

บทที่ 3 วิธีดำเนินงาน

3.1 ขั้นตอนการดำเนินงาน

- 3.1.1 คิดหัวข้อโครงงาน เพื่อนำเสนออาจารย์ที่ปรึกษา
- 3.1.2 ศึกษาค้นคว้า รวบรวมข้อมูลเกี่ยวกับการสร้าง SignGlove
- 3.1.3 ศึกษาการใช้โปรแกรมที่เกี่ยวกับการพัฒนา SignGlove
- 3.1.4 จัดทำแผนโครงงาน เพื่อนำเสนออาจารย์ที่ปรึกษา
- 3.1.5 ออกแบบอุปกรณ์ การทำงาน
- 3.1.6 นำไปปรึกษาอาจารย์ที่ปรึกษา ว่ามีส่วนไหนควรปรับแก้หรือไม่
- 3.1.7 จัดทำ SignGlove และทดลองใช้จริง
- 3.1.8 ตรวจสอบและแก้ไขปัญหาของระบบการทำงาน
- 3.1.9 ทำเอกสารสรุประยงานโครงงาน

3.2 วิธีการรวบรวมข้อมูล

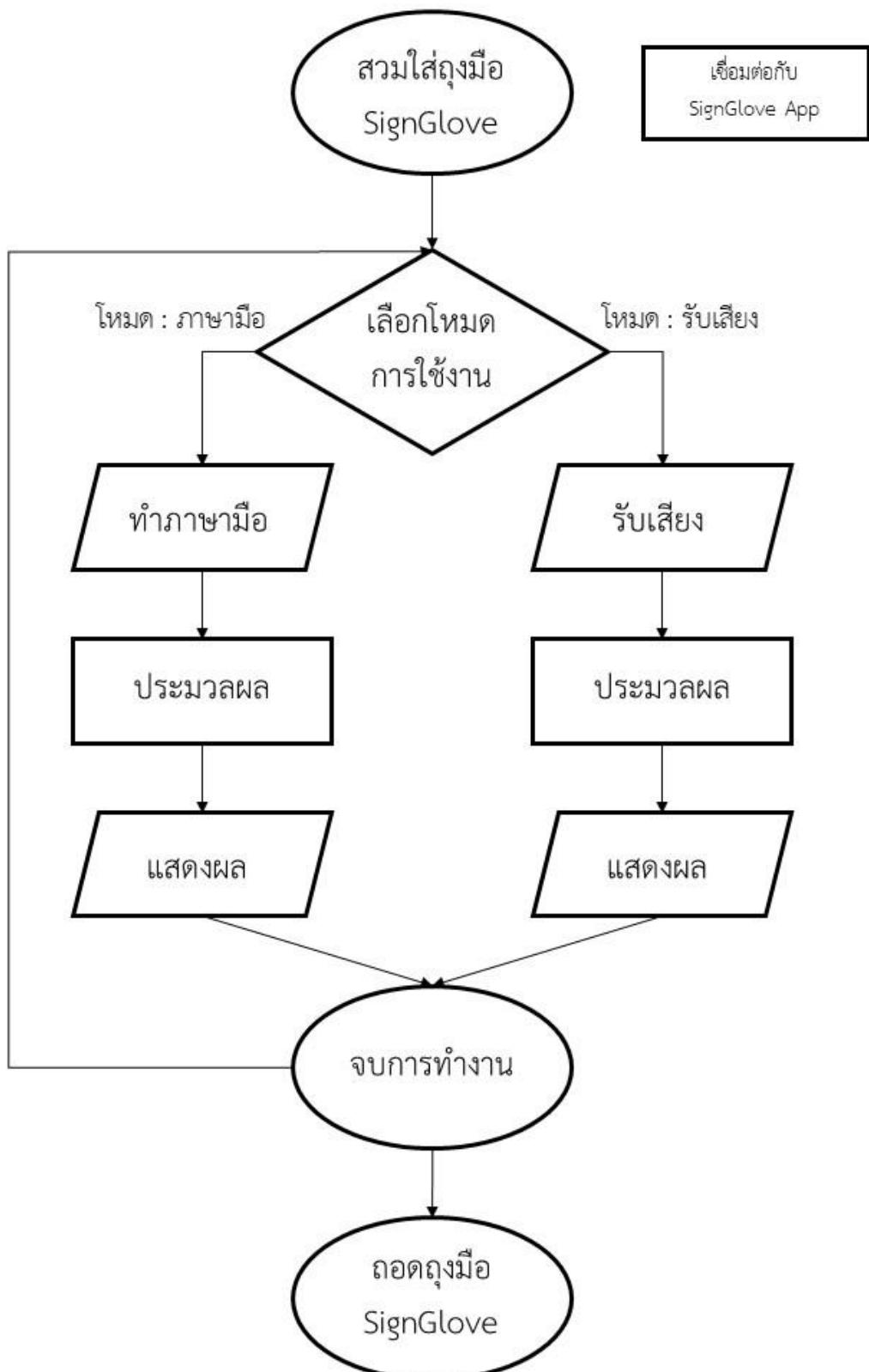
- 3.2.1 ศึกษา เรื่อง บุคคลที่มีความบกพร่องทางการได้ยิน
- 3.2.2 ศึกษา เรื่อง บุคคลที่มีความบกพร่องทางการพูดและภาษา
- 3.2.3 ศึกษา เรื่อง การเขียน code ภาษา C/C++
- 3.2.4 ศึกษา เรื่อง การใช้ Arduino

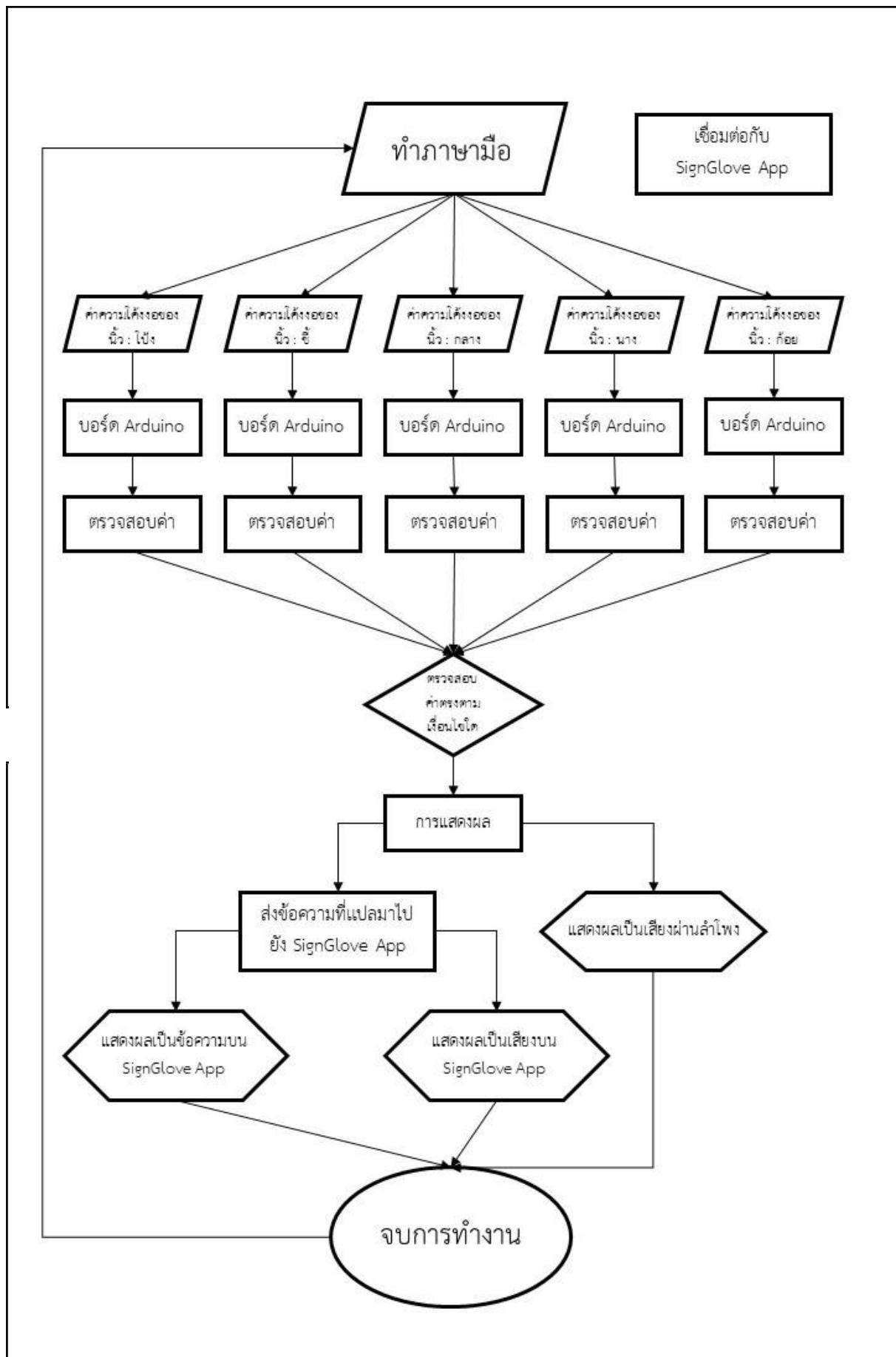
3.3 อุปกรณ์และเครื่องมือที่ใช้

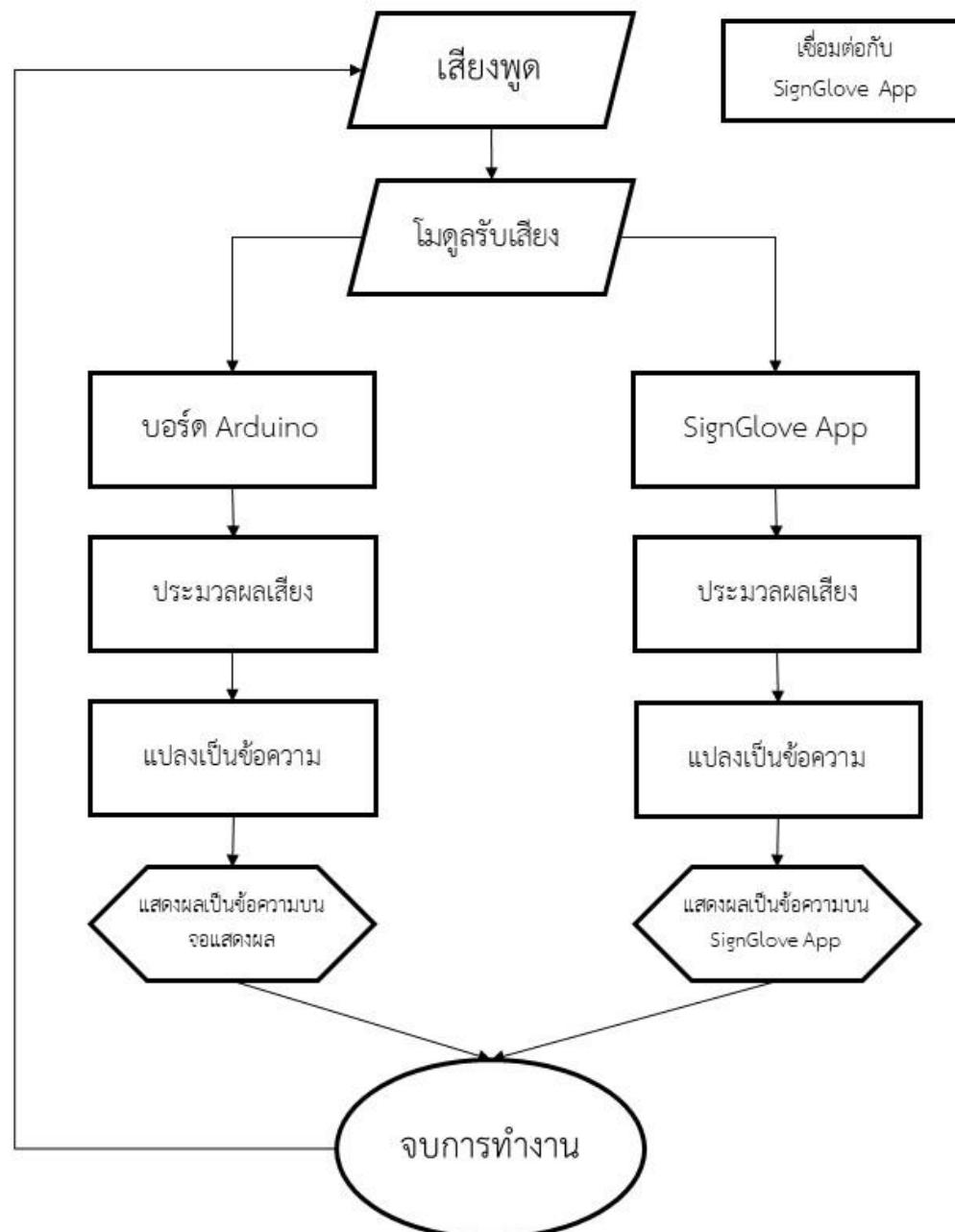
- 3.3.1 โปรแกรมที่ใช้ Arduino IDE, visual studio, visual code, Blender, Sketchup ในการทำงาน แต่ละโปรแกรมมีหน้าที่ดังนี้
 - Arduino IDE ใช้ในการเขียนโค้ดระบบการทำงาน
 - visual studio, visual code, ในการสร้างแอปแสดงผลการทำงาน
 - Blender, Sketchup ในการสร้างโมเดลสามมิติ
- 3.3.2. เครื่องมือที่ใช้แสดงผล โทรศัพท์มือถือ และ จอLED

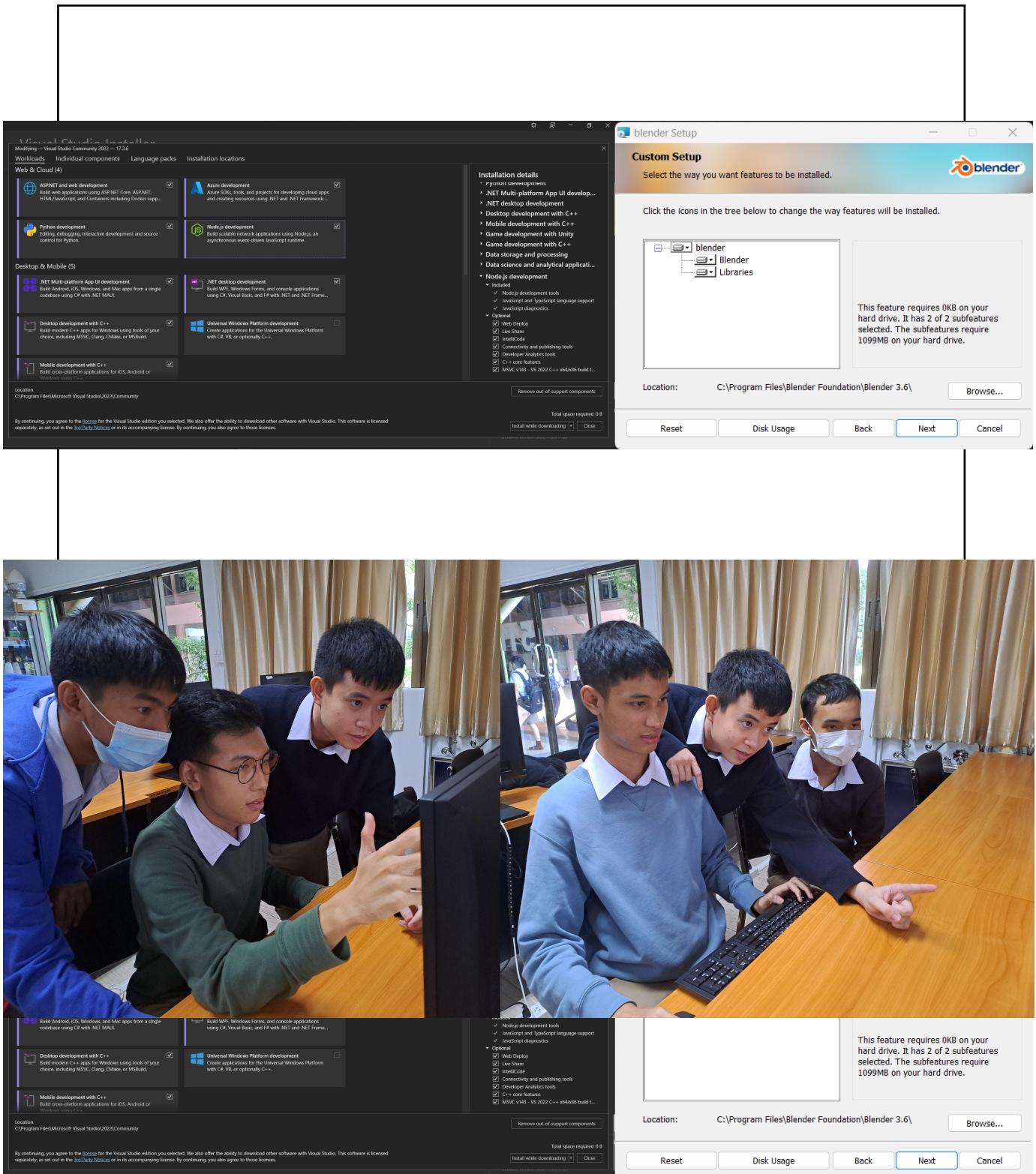
3.3.3 อุปกรณ์ใช้ทำ โทรศัพท์มือถือ และ คอมพิวเตอร์ที่ดังนี้

- บอร์ด Arduino (เช่น Arduino Uno หรือ Arduino Nano)

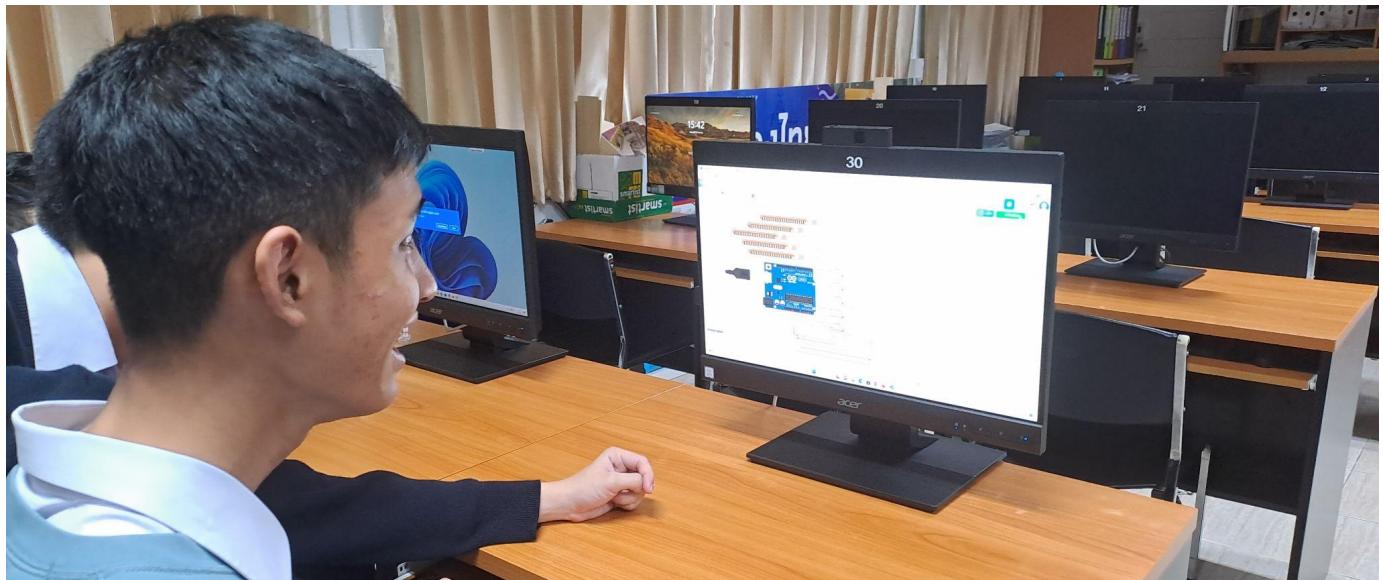
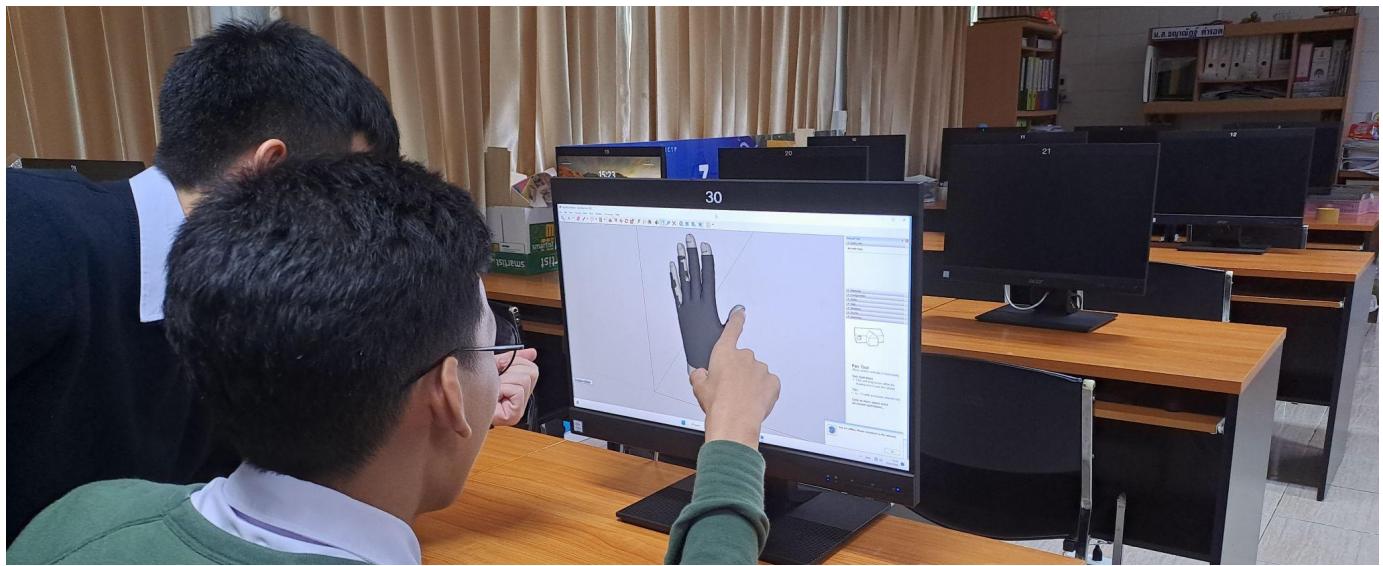


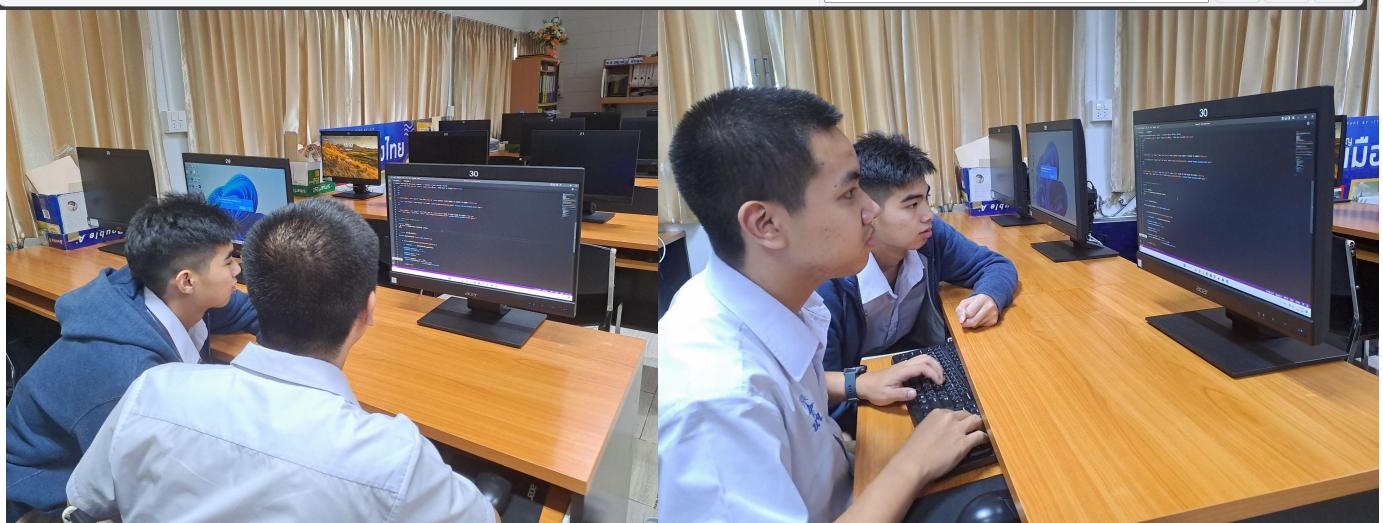
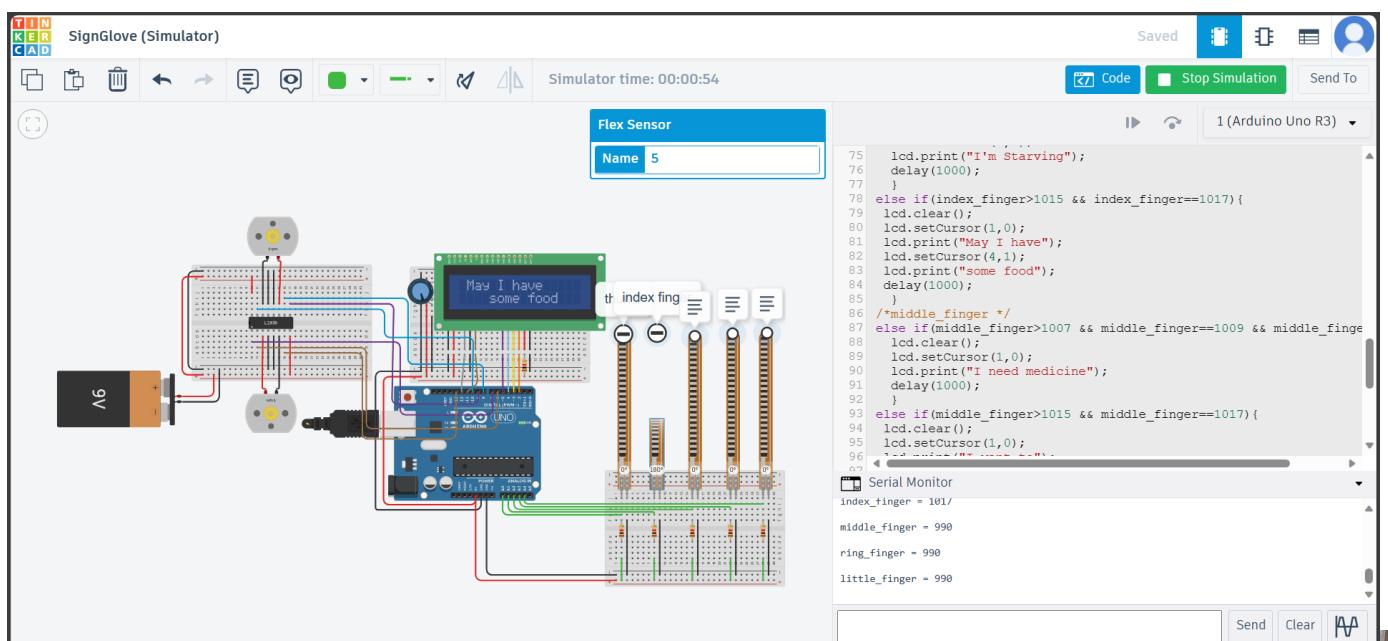






3.5.2 ศึกษาการใช้โปรแกรม Arduino IDE, visual studio, visual code, Blender, Sketchup





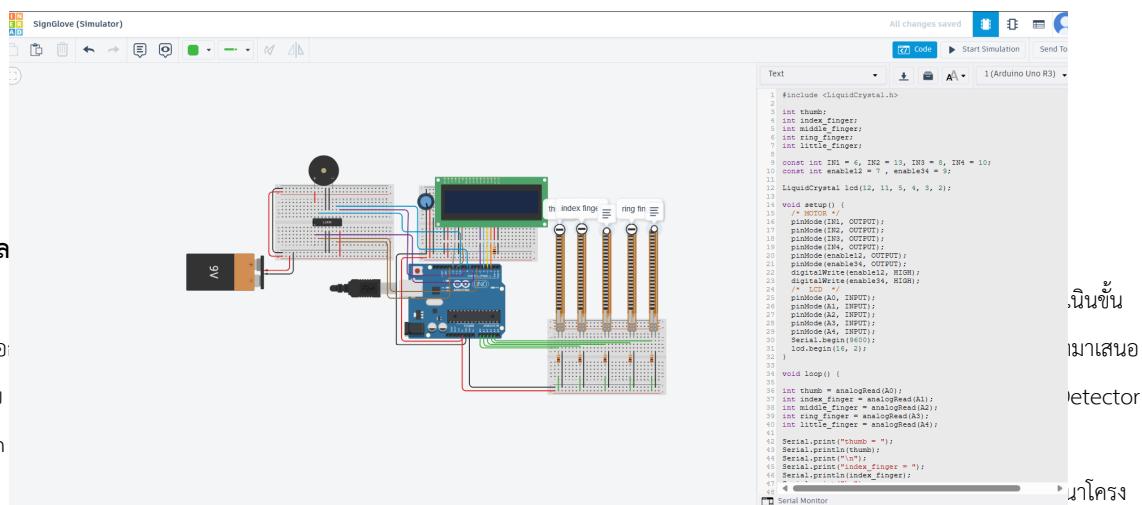
3.5.9 จัดเตรียมซื้ออุปกรณ์ ตามรายการกำหนด และทำตามที่อุปกรณ์

*แต่เนื่องจากปัจจุบัน ทุน ในการจัดทำนั้นสูงเกินไป ถึงข้างละ 2570 บาท ทำให้ทางผู้จัดทำได้จัดทำเป็น simulator แทน

บัญชีบประมาณ

4.1 ผล

ตอน ออ
ผ่านเว็บ
Python



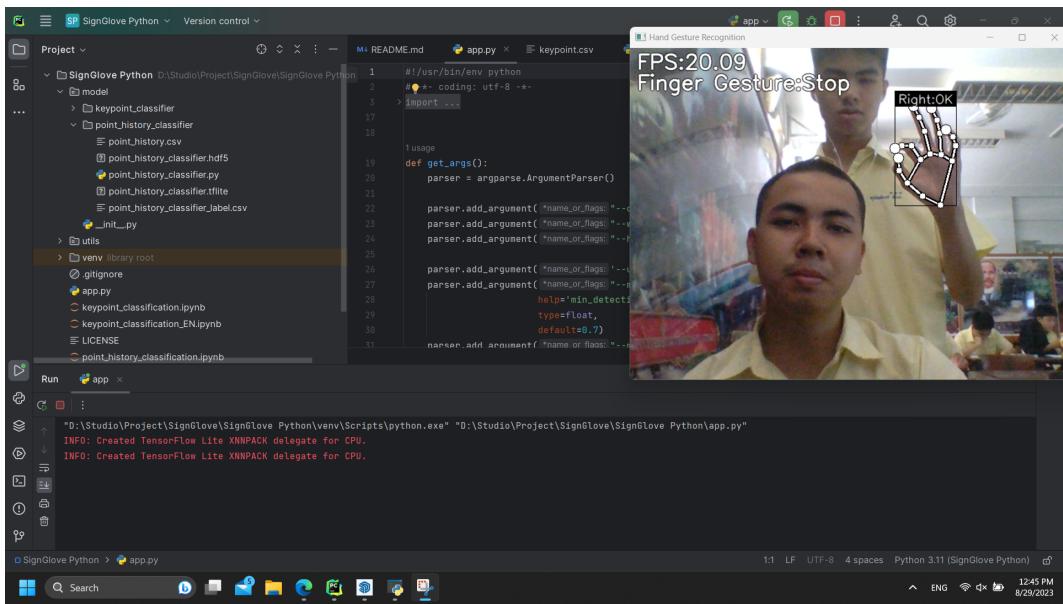
งานและได้ฝึกทักษะการใช้เทคโนโลยีสร้างผลงานที่เป็นประโยชน์

4.2 ตัวอย่างผลงาน

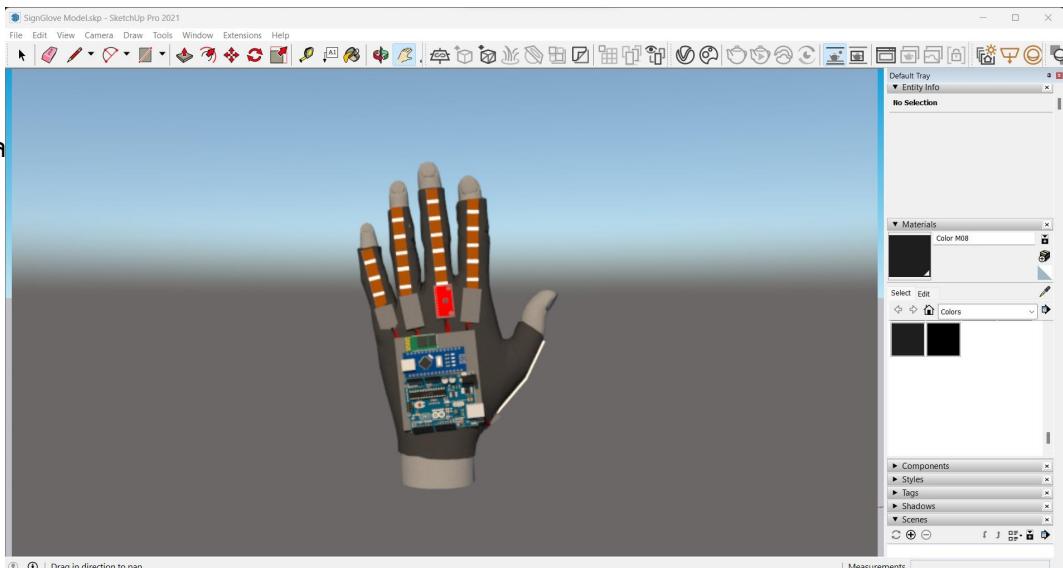
4.2.1 แบบจำลองการทำอุปกรณ์ SignGlove ด้วยเว็บ www.tinkercad.com

ราคารวมทั้งหมด

฿2,570.00



4.2.3 แบบจำลองการทำงานของอุปกรณ์



4.3 ผล

สถานภาพ	จำนวนผู้ทดลอง (n = 2)	
	จำนวน	ร้อยละ
ผู้มีสถานภาพปกติ	2	100
ผู้มีสถานภาพพิการ	0	0

4.3.1.1 จากตารางพบว่า ผู้ทดลองใช้หั้งหมวด 2 คน เป็น ผู้มีสถานภาพปกติ คิดเป็นร้อยละ 100 และ ผู้มีสถานภาพพิการ คิดเป็นร้อยละ 0

4.3.2 ผลการประเมิน SignGlove ในรูปแบบ simulator

ผลจากการประเมินในการทดลองใช้ SignGlove ในรูปแบบ simulator โดยกำหนด เกณฑ์คะแนนค่าเฉลี่ย ดังนี้

1.00 – 1.49 หมายถึง มีความพึงพอใจ อยู่ในระดับ น้อยที่สุด

1.50 – 2.49 หมายถึง มีความพึงพอใจ อยู่ในระดับ น้อย

2.50 – 3.49 หมายถึง มีความพึงพอใจ อยู่ในระดับ ปานกลาง

3.50 – 4.49 หมายถึง มีความพึงพอใจ อยู่ในระดับ หากที่สุด

4.50 – 5.00 หมายถึง มีความพึงพอใจ อยู่ในระดับ หากที่สุด

หัวข้อประเมิน	ระดับคะแนน	
	คะแนนเฉลี่ย	ความพึงพอใจ
ความน่าสนใจของโครงงาน	5	มากที่สุด
ความสะดวกในการใช้งานอุปกรณ์ SignGlove	3.75	มาก
ความถูกต้องในการแปลภาษาไม้อ	3	ปานกลาง
ความถูกต้องในการแปลเสียง	4.25	มาก
ขีดความสามารถในการแปล	3.5	มาก
ความสะดวกในการใช้งาน SignGlove App	4	มาก
การจัดวางองค์ประกอบภายใน SignGlove App อย่างเหมาะสม	3.25	ปานกลาง

4.3.2.1 จากตาราง การประเมิน พบร่วมกับความพึงพอใจในการใช้ในการใช้ในการใช้ข้อมูลของโครงงาน โดยรวมอยู่ในระดับ มาก เมื่อพิจารณาเป็นรายข้อ พบว่า มีระดับความพึงพอใจมากที่สุด คือ ความน่าสนใจของโครงงาน

4.3.2.2 จากตาราง การประเมิน พบร่วมกับความพึงพอใจร่วมอยู่ที่ปัญหาด้าน ความถูกต้องในการแปลภาษาเมื่อ ที่มีความถูกต้อง ชัดเจนน้อยอยู่ในระดับ ปานกลาง และ การจัดวางองค์ประกอบภายใน SignGlove App ที่ยังไม่เหมาะสมเท่าไหร่นักทำให้ใช้งานได้ยาก

บทที่ 5 สรุปผลการดำเนินงาน

5.1 การดำเนินงานจัดทำโครงงาน

5.1.1 วัตถุประสงค์

5.1.1.1 เพื่อศึกษาเกี่ยวกับความบกพร่องทางการได้ยิน, การพูด และค้นคว้าหาทางช่วยให้สามารถสื่อสารอย่างปากต่อปาก

5.1.1.2 เพื่อศึกษาการใช้ภาษาเมื่อ

5.1.1.3 เพื่อพัฒนาผู้เรียนให้มีความสามารถทาง การคิด, ตรรกะ เพิ่มขึ้นเนื่องจากการเขียนการทำงานของระบบ ด้วยภาษาC/C++ และบอร์ด arduino เป็นสิ่งที่เคยศึกษามาก่อนจึงอยากศึกษาเพิ่มขึ้นซึ่งเป็นการพัฒนาตนเอง

5.1.1.3 เพื่อพัฒนาผู้เรียนให้มีความสามารถทาง การคิด, ตรรกะ เพิ่มขึ้นเนื่องจากการเขียนการทำงานของระบบ ด้วยภาษาpython ซึ่งเป็นสิ่งที่เคยศึกษามาก่อนจึงอยากศึกษาเพิ่มขึ้นซึ่งเป็นการพัฒนาตนเองและ opencv ที่เป็นสิ่งใหม่น่าสนใจ

5.1.2 วัสดุอุปกรณ์และเครื่องมือที่ใช้

5.1.2.1 โปรแกรมที่ใช้ Arduino IDE, visual studio, PyCharm, Sketchup ในการทำงาน แต่ละโปรแกรมมีหน้าที่ดังนี้

- Arduino IDE ใช้ในการเขียนโค้ดระบบการทำงาน
- PyCharm ใช้ในการเขียนโค้ดระบบการทำงานของแบบจำลอง
- visual studio ในการสร้างแอปแสดงผลการทำงาน
- Sketchup ในการสร้างโมเดลสามมิติ

5.2 ผลที่ได้รับ ปัญญา และ อุปสรรค

5.2.1 ในส่วนของแบบจำลอง ช่วยทำให้ผู้ที่สนใจได้ตัวโครงงาน SignGlove สามารถเข้าใจวิธีการทำงานของ SignGlove ได้เพื่อทดสอบ ในส่วนของการทำขึ้นงานจริง ด้วยแบบจำลองภายในเว็บ www.tinkercad.com, โน๊ตสมมติ และ แบบจำลองของ OpenCV ในหารตรวจสอบ มือและแปลความหมายตามที่ตรวจจับได้

ในส่วนของคนจะจัดทำ โครงการนี้ทำให้คนจะจัดทำได้เรียนรู้ขั้นตอนการพัฒนาโปรแกรมเกณฑ์สามารถตรวจสอบจับการเคลื่อนไหวของมือได้ ซึ่งเป็นสิ่งที่ท้าทายอย่างมากเนื่องจากเป็นการแก้ปัญหาเฉพาะหน้าที่ไม่คาดคิดมาก่อนว่าจะทำคนจะจัดทำสามารถเพิ่มทักษะในการออกแบบโปรแกรมเชิงวัตถุและพัฒนาโปรแกรมโดยใช้ภาษา C/C++ ร่วมกับ Arduino IDE, PyCharm ประกอบกับการทำโมเดลสามมิติร่วมกับ Sketchup เป็นการนำความรู้ที่ได้เรียนมาใช้ให้เกิดประโยชน์เพื่อสังคม

5.2.2 หลังจากได้ดำเนินการออกแบบอุปกรณ์แล้ว ทางคนจะจัดทำจึงได้ทำการสรุปงบประมาณแล้วพบว่า งบประมาณที่ใช้ทำโครงการนั้นสูงเกินไปจึงได้จัดทำเป็น แบบจำลองแทน

5.2.3 พบว่ามีปัญหาของโครงการโดยร่วมอยู่ที่ปัญหาด้าน ความถูกต้องในการแปลงภาษาเมื่อ ที่มีความถูกต้องชัดเจนน้อยอยู่ในระดับปานกลาง และ การจัดวางองค์ประกอบภายใน SignGlove App ที่ยังไม่เหมาะสมเท่าไหร่นักทำให้ใช้งานได้ยาก

5.3 ข้อเสนอแนะต่อโครงการ

5.3.1. เป็นโครงการที่ต้นทุนไม่มาก ถ้าหากสามารถทำตัวอุปกรณ์ได้จริง และสามารถนำช่วยเหลือผู้พิการ ในการสื่อสารกับผู้อื่น มีการเข้าถึงง่าย สะดวก และรวดเร็ว จะดีมาก ควรพัฒนาต่อไป ให้สามารถใช้งานอย่างทั่วถึง

5.3.2. ออกแบบอุปกรณ์ ให้ความกระหัดลัดมากกว่านี้ให้ดูดี และ แอพพลิเคชันออกแบบใหม่มีการใช้งานได้ง่ายกว่าองค์ประกอบดูทันสมัย

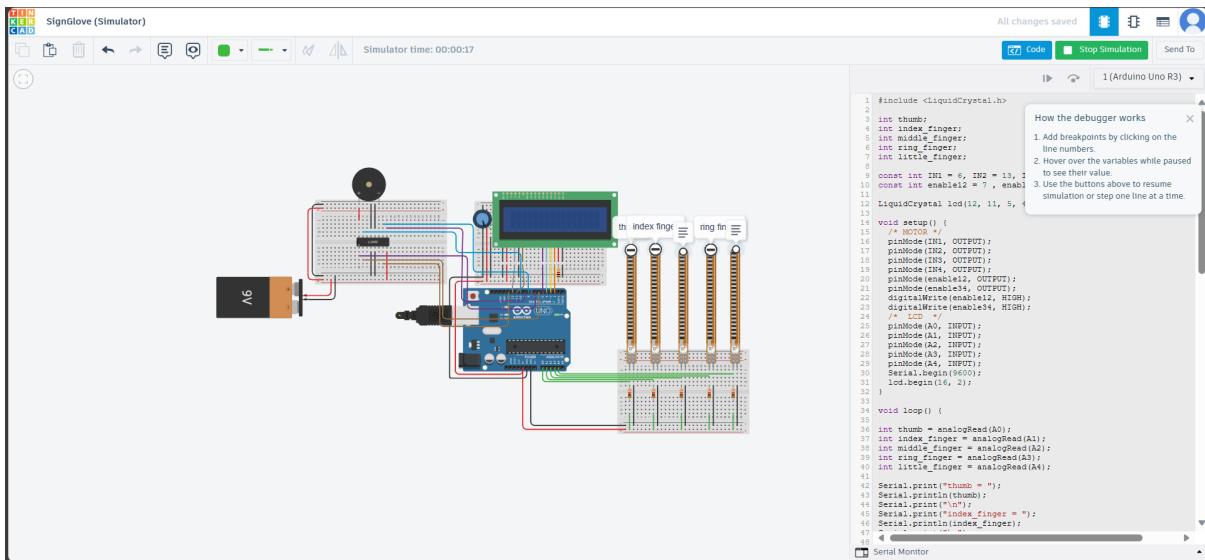
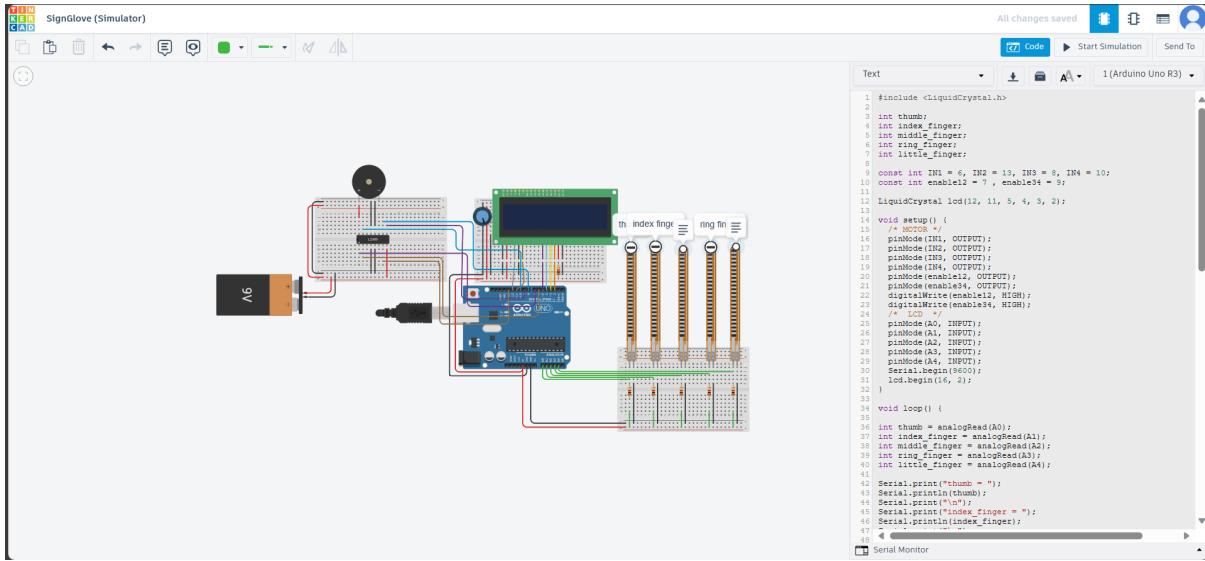
5.3.3 พัฒนาระบบที่ประสานการทำงานระหว่างอุปกรณ์ และ แอพพลิเคชันให้สามารถมีความหลากหลายในการใช้งานและเชื่อมต่อ ง่ายกว่านี้

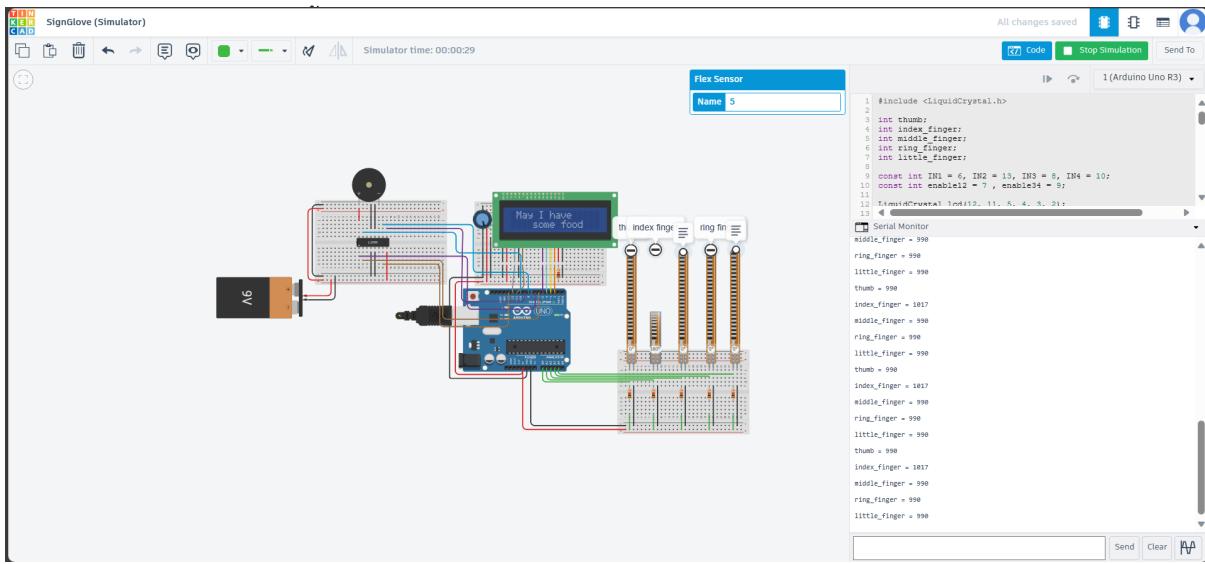
5.4 สรุปผลการดำเนินงานโครงการ

การพัฒนาโครงการ SignGlove(Hand sign language translator glove) นี้ ผู้จัดทำได้เริ่มดำเนินงานตามขั้นตอนการดำเนินงานที่เสนอในบทที่ 3 และ แล้วได้พบว่ามีปัญหาในเรื่องของปัจจัยทุนการสร้างทำให้ไม่สามารถจัดทำตัวขึ้นจริง ออกแบบได้ ทางผู้จัดทำจึงได้ทำการจัดในส่วนของแบบจำลองการใช้งานขั้นมาแทน เพื่อเป็นแนวทางในการศึกษาและนำไปต่อยอดทำเป็นตัวขึ้นงานจริงได้ในภายหลังโดยง่าย ซึ่งได้นำตัวแบบจำลองการใช้งานมาเผยแพร่ที่เว็บ GitHub ในชื่อ [junyawat1019 \(Daryl\) \(@github.com\)](#) ซึ่งสามารถเข้าไปโหลดตัวแบบจำลองการใช้งานและ รายงานโครงการนี้ได้ สามารถนำไปใช้เป็นตัวอย่างในการจัดทำโครงการอื่นต่อไป เพื่อเป็นประโยชน์ใน การพัฒนาสังคม และเทคโนโลยี เพื่อผู้พิการจะได้สามารถใช้ชีวิตได้อย่างภาคภูมิ โดยการที่โครงการเป็นไปได้ด้วยดี ต้องขอบคุณทั้งครูที่ปรึกษาเพื่อนๆ ในห้องเรียนได้เข้าไปมีส่วนร่วมในการ จัดการเรียนรู้ โดยแสดงความเห็นในเนื้อหาและรูปแบบของการนำเสนออย่างหลากหลาย ซึ่งทำให้เกิดการเรียนรู้และการพัฒนาสืบต่อไป

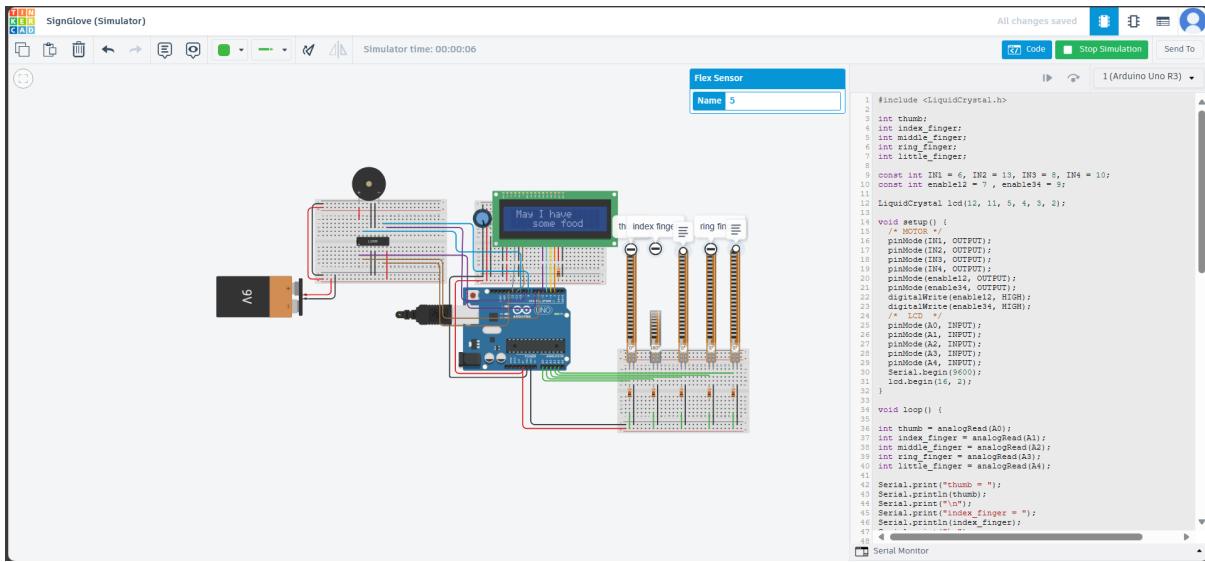
ภาคผนวก

ก.แบบจำลองด้วยเว็บ www.tinkercad.com





4. เมื่อปรับค่าความต้องของนิ้วตรงตามหลักภาษาไม้อแล้ว จะแสดงความหมายทันที



ข.แบบจำลองด้วยเว็บ OpenCV

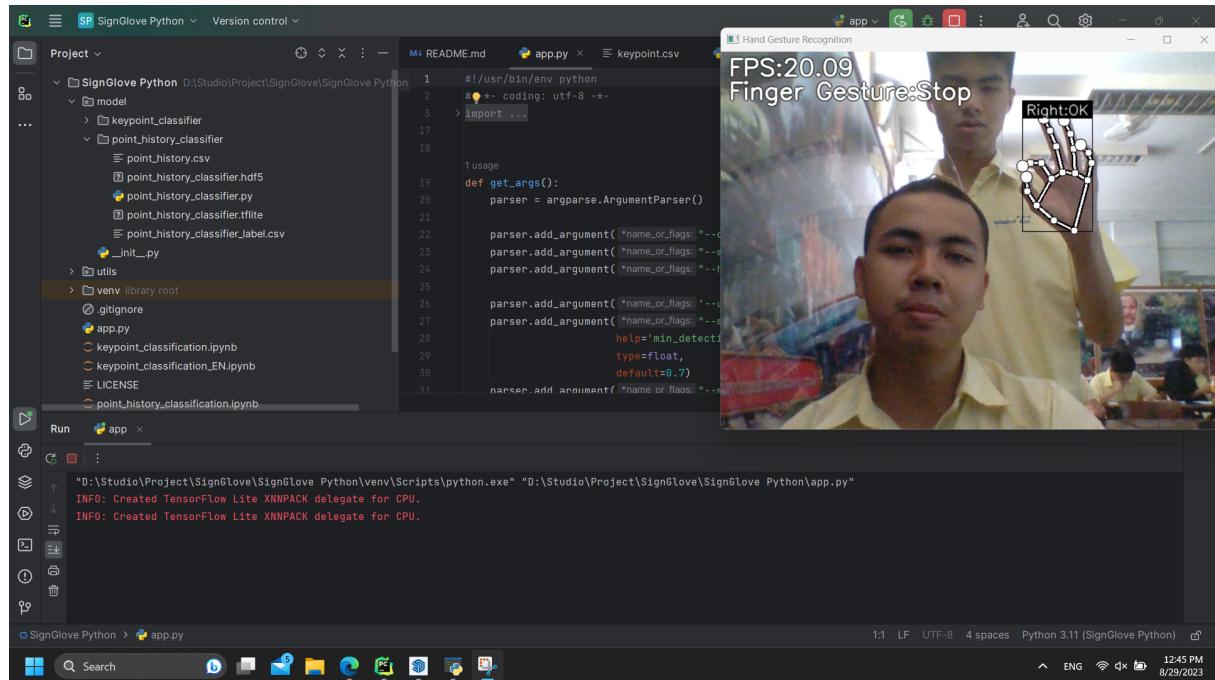
1. ค่ากรับเคลื่อนไหวนิวท์เกล่องจับการเคลื่อนไหวได้

The screenshot shows the PyCharm interface with the project 'SignGlove Python' open. A warning message at the top states: "The file size (3.65 MB) exceeds the configured limit (2.56 MB). Code insight features are not available." The main view displays a large CSV file named 'keypoint_classifier_label.csv' containing 29 columns of numerical data. The columns are labeled from 1 to 29, representing various keypoint classification values.

2. นำค่าที่ได้มาแทนเพื่อให้โปรแกรมรู้ความหมายของภาษาเมือง

The screenshot shows the PyCharm interface with the project 'SignGlove Python' open. The 'keypoint_classifier_label.csv' file is selected in the project tree. A notification on the right side of the interface states: "Microsoft Defender con... The IDE has detected Microsoft Defender with Real-Time Protection enabled. It might severely degrade IDE performance. It is recommended to add following paths to the Defender folder exclusion list: C:\Users\Tangl\AppData\Ro...". The file contains 5 rows of data: Open, Close, Pointer, OK, and an empty row.

3. เมื่อตัวโปรแกรมรู้ความหมายแล้วว่าหากเรากดเริ่มโปรแกรมและแสดงภาษาไทยให้กล้องเห็นจะมีการแปลภาษาที่มือออก มาทันที



ค. โนมเดลสามมิติ

1. โนมเดลสามมิติ SignGlove (ด้านหน้า)



2. โมเดลสามมิติ SignGlove (ด้านขวา)



3. โมเดลสามมิติ SignGlove (ด้านซ้าย)

