



ชุดคำสั่งหมวดเซนเซอร์ตรวจจับสิ่งกีดขวางสำหรับผู้พิการทางสายตา

INSTRUCTION SET CAP FOR VISUAL IMPAIRMENT

จัดทำโดย

นายจิราเมธ ประเทือง

นายพลศวัจน์ คำคง

นายพรคนัย โอษฐ์ประไพ

โครงการนี้เป็นส่วนหนึ่งของวิชาโครงการ หลักสูตรประกาศนียบัตรวิชาชีพ

สาขาวิชาเทคโนโลยีสารสนเทศ

วิทยาลัยเทคโนโลยีอรรถวิทย์พัฒนการ

ปีการศึกษา 2562

ชุดคำสั่งหมวดเซนเซอร์ตรวจจับสิ่งกีดขวางสำหรับผู้พิการทางสายตา

**INSTRUCTION SET CAP VISUAL IMPAIRMENT**

จัดทำโดย

นายจิราเมธ ประเทือง

นายพลศวัจน์ คำคง

นายพรคนัย โอษฐ์ประไพ

โครงการนี้เป็นส่วนหนึ่งของวิชาโครงการ หลักสูตรประกาศนียบัตรวิชาชีพ

สาขาวิชาเทคโนโลยีสารสนเทศ

วิทยาลัยเทคโนโลยีอรรณพวิทยพัฒน์

ปีการศึกษา 2562

**COPYRIGHT 2019**

**COLLEGE OF INFORMATION TECHNOLOGY**

**ATTAWIT COMMERCIAL TECHNOLOGY COLLEGE**

**COPYRIGHT 2019**

**COLLEGE OF INFORMATION TECHNOLOGY**

**ATTAWIT COMMERCIAL TECHNOLOGY COLLEGE**



ชื่อโครงการภาษาไทย

ชุดคำสั่งหมวดเซนเซอร์ตรวจจับสิ่งกีดขวางสำหรับผู้พิการทางสายตา

ชื่อโครงการภาษาอังกฤษ

Instruction Set Cap for Visual Impairment

|     |               |            |                    |
|-----|---------------|------------|--------------------|
| โดย | 1.นายจิราเมธ  | ประเทือง   | รหัสประจำตัว 39344 |
|     | 2.นายพลศวัจน์ | คำคง       | รหัสประจำตัว 39476 |
|     | 3.นายพรคนัย   | โอษฐ์ประไพ | รหัสประจำตัว 39658 |

คณะกรรมการอนุมัติให้เอกสารโครงการฉบับนี้เป็นส่วนหนึ่งของการศึกษา วิชาโครงการตามหลักสูตรประกาศนียบัตรวิชาชีพ (ปวช.) สาขาวิชาเทคโนโลยีสารสนเทศ วิทยาลัยเทคโนโลยีอรรถวิทย์พัฒนศึกษา (ATC)

.....  
(อาจารย์คุณานนท์ สุขเกษม)

อาจารย์ที่ปรึกษา

.....  
(อาจารย์สุธารัตน์ ทองใหม่)

อาจารย์ที่ปรึกษา

.....  
(อาจารย์คุณานนท์ สุขเกษม)

หัวหน้าสาขาวิชาเทคโนโลยีสารสนเทศ

## บทคัดย่อ

|                  |   |              |       |
|------------------|---|--------------|-------|
| หัวข้อโครงการ    | ชุดคำสั่งหมวกเซนเซอร์ตรวจจับสิ่งกีดขวางสำหรับผู้พิการทางสายตา |              |       |
| ผู้จัดทำโครงการ  | 1.นายจิราเมธ ประเทือง   | รหัสประจำตัว | 39344 |
|                  | 2.นายพลสวัจน์ คำคง  | รหัสประจำตัว | 39476 |
|                  | 3.นายพชรดนัย โอษฐ์ประไพ                                       | รหัสประจำตัว | 39658 |
| อาจารย์ที่ปรึกษา | 1.อาจารย์คุณานนท์ สุขเกษม                                     |              |       |
|                  | 2.อาจารย์สุธารัตน์ ทองใหม่                                    |              |       |
| สาขาวิชา         | เทคโนโลยีสารสนเทศ   |              |       |
| สถาบัน           | วิทยาลัยเทคโนโลยีอรรถวิทย์พาณิชการ ปีการศึกษา 2562            |              |       |

---

## บทคัดย่อ

ชุดคำสั่งหมวกเซนเซอร์ตรวจจับสิ่งกีดขวางสำหรับผู้พิการทางสายตา สร้างเพื่อเตือนผู้พิการทางสายตาไม่ให้ผู้พิการทางสายตาเกิดอันตรายในชีวิตประจำวัน สะดวกต่อการใช้ชีวิต โดยใช้หมวกเซนเซอร์เพื่อป้องกันการเกิดอุบัติเหตุ ต่อผู้พิการทางสายตาสถานต่างๆ

การใช้งานหมวกสามารถเตือนได้โดยใช้ Buzzer หรือเสียงสัญญาณเตือนสำหรับ Arduino และมีบอร์ดควบคุมการทำงานเปรียบเสมือน ตัวควบคุมการทำงานของอุปกรณ์ภายในหมวกเซนเซอร์ Ultrasonic ตัวแรกจะตรวจภายในระยะ 1 เมตร และตัวที่ประมาณ 1.5 เมตร ในระดับล่างมี Switch ควบคุมการเปิดและปิดของตัวขึ้นงาน หมวกสามารถพกพามีขนาดที่เหมาะสมกับการถือหรือสวมไปที่อื่น

โครงการนี้มีประโยชน์ต่อผู้พิการทางสายตาเป็นอย่างมากและผู้ที่เข้าถ้ำในช่วงดึกที่มองไม่เห็นและช่วยป้องกันอันตรายที่อยู่ในป่าหรือถ้ำได้ ต่อผู้พิการทางสายตาสามารถเตือนเมื่อเจอสิ่งของที่ไม่นเห็น การพัฒนาได้โดยการนำเซนเซอร์ที่สามารถตรวจสอบหลุม และยังพัฒนาให้เกิดความเสถียรมากขึ้น

## กิตติกรรมประกาศ

โครงการฉบับนี้สำเร็จลุล่วงได้อย่างดีก็เพราะได้รับกรุณาจากอาจารย์ คุณานนท์ สุขเกษม และอาจารย์ สุรรัตน์ ทองใหม่ ที่ได้ให้คำแนะนำและคำปรึกษาเกี่ยวกับการเขียนชุดคำสั่งหมวด เซนเซอร์ตรวจจับสิ่งกีดขวางสำหรับผู้พิการทางสายตา ผู้ทำโครงการรู้สึกซาบซึ้งในความอนุเคราะห์จากท่านเป็นอย่างมาก และกราบขอบพระคุณเป็นอย่างสูง

ผู้ทำโครงการขอกราบขอบพระคุณ คุณพ่อ คุณแม่ และเพื่อนๆ พี่ๆ ที่ให้กำลังใจและให้โอกาสบุตรได้รับการศึกษาในระดับต่างๆ จนกระทั่งได้สำเร็จการศึกษาระดับประกาศนียบัตรวิชาชีพนี้รวมทั้งคณาจารย์ทุกท่านในสาขาวิชาเทคโนโลยีสารสนเทศที่ได้ประสิทธิ์ประสาทวิชาความรู้และคำสั่งสอนให้กับผู้ทำโครงการในการเรียนทุกระดับชั้น

ขอขอบพระคุณครอบครัวที่ให้การช่วยเหลือในทุกๆ ด้าน จนทำให้ประสบความสำเร็จ ลุล่วงไปได้ด้วยดี

ขอขอบคุณ พี่น้องๆ ทุกคนในสาขาวิชาเทคโนโลยีสารสนเทศ วิทยาลัยเทคโนโลยีอรรณพวิทยพัฒน์วิชาการ ที่เป็นกำลังใจและคอยให้ความช่วยเหลือในการทำโครงการฉบับนี้

สุดท้ายความรู้และประโยชน์ที่ได้รับจากโครงการฉบับนี้ผู้ทำโครงการขอมอบความดีที่ได้นี้ให้แก่ผู้มีพระคุณทุกท่าน

นาย จิราเมธ ประเทือง

นาย พลศวัจน์ คำคง

นาย พชรดนัย โอบรู้ประไพ

## สารบัญ

หน้า

|   |     |
|---|-----|
| บทคัดย่อภาษาไทย .....                                   | I   |
| กิตติกรรมประกาศ.....                                    | II  |
| สารบัญ .....  | III |
| สารบัญ(ต่อ).....  | IV  |
| สารบัญตาราง .....                                       | V   |
| สารบัญรูป .....   | VI  |
| บทที่ 1 บทนำ.....                                       | 1   |
| 1.1 ความเป็นมาและความสำคัญของปัญหา .....                | 1   |
| 1.2 วัตถุประสงค์ของโครงการ .....                        | 1   |
| 1.3 ขอบเขตของโครงการ .....                              | 2   |
| 1.4 ขั้นตอนการดำเนินงาน .....                           | 2   |
| 1.5 ระยะเวลาการทำโครงการ .....                          | 5   |
| 1.6 ประโยชน์ที่คาดว่าจะได้รับ .....                     | 7   |
| 1.7 งบประมาณที่ใช้ในการทำโครงการ .....                  | 7   |
| บทที่ 2 เอกสารและทฤษฎีที่เกี่ยวข้อง .....               | 8   |
| 2.1 ความต้องการของระบบที่เหมาะสม.....                   | 8   |
| 2.2 สอนใช้งาน Arduino ควบคุม ultrasonic วัดระยะทาง..... | 8   |
| 2.3 ภาษาซี (C Programing Language).....                 | 9   |
| 2.4 การใช้ if else.....                                 | 31  |
| 2.5 การคำนวณระยะทาง .....                               | 46  |
| 2.6 วิธีการติดตั้ง Driver Arduino .....                 | 51  |
| บทที่ 3 วิธีดำเนินโครงการ.....                          | 60  |
| 3.1 วางแผนและเตรียมการและเตรียมอุปกรณ์.....             | 60  |
| 3.2 ระบบการทำงานของชิ้นงาน .....                        | 65  |
| 3.3 สร้างชุดคำสั่ง .....                                | 66  |
| บทที่ 4 ผลการศึกษา .....                                | 68  |
| 4.1 ขั้นตอนการออกแบบชิ้นงาน .....                       | 68  |
| 4.2 ขั้นตอนการทำงาน .....                               | 68  |
| 4.3 ขั้นตอนการทดสอบ .....                               | 69  |



## สารบัญ(ต่อ)

|   |           |
|---|-----------|
| 4.4 ผลการทดลองโดยใช้โค้ดคำสั่ง.....             | 69        |
| <b>บทที่ 5 สรุป อภิปราย และข้อเสนอแนะ .....</b> | <b>71</b> |
| 5.1 วัตถุประสงค์ของโครงการ.....                 | 71        |
| 5.2 ประโยชน์ที่ได้รับจากการศึกษา.....           | 71        |
| 5.3 ปัญหาที่ประสบในการดำเนินโครงการ .....       | 72        |
| 5.4 ผลการดำเนินโครงการ.....                     | 72        |
| 5.5 อภิปราย .....                               | 72        |
| 5.6 ข้อเสนอแนะ .....                            | 72        |
| <b>บรรณานุกรม.....</b>                          | <b>73</b> |
| <b>ภาคผนวก.....</b>                             | <b>74</b> |
| <b>ภาคผนวก ก.....</b>                           | <b>75</b> |
| แบบเสนอร่างโครงการ .....                        | 76        |
| <b>ภาคผนวก ข.....</b>                           | <b>84</b> |
| ความคืบหน้า.....                                | 85        |
| <b>ภาคผนวก ค.....</b>                           | <b>94</b> |
| คู่มือการใช้งาน.....                            | 95        |
| <b>ภาคผนวก ง.....</b>                           | <b>97</b> |
| ประวัติผู้จัดทำโครงการ .....                    | 98        |

## สารบัญตาราง

| ตารางที่                          | หน้า |
|-----------------------------------|------|
| 1.1 ตารางแผนการดำเนินงาน .....    | 5    |
| 3.1 แสดงแผนการดำเนินโครงการ ..... | 61   |

## สารบัญรูป

| รูปที่   | หน้า |
|--|------|
| 1.1 แผนภาพ Flowchart ขั้นตอนการดำเนินงาน .....                         | 3    |
| 1.1 แผนภาพ Flowchart ขั้นตอนการดำเนินงาน (ต่อ) .....                   | 4    |
| 2.1 ตัวอย่างชุดคำสั่ง ultrasonic sensor .....                          | 9    |
| 2.2 Timing Diagram ของเซ็นเซอร์วัดระยะด้วยคลื่นอัลตราโซนิก HCSR04..... | 47   |
| 2.3 Download The Arduino Software .....                                | 51   |
| 2.4 Download Driver Arduino 1.0.1 .....                                | 52   |
| 2.5 Driver Arduino Zip .....   | 52   |
| 2.6 Manage Computer.....   | 53   |
| 2.7 System.....  | 53   |
| 2.8 Device Manager 1 .....   | 54   |
| 2.9 Device Manager 2 .....   | 55   |
| 2.10 Update Driver Software 1 .....                                    | 55   |
| 2.11 Search For Driver Software .....                                  | 56   |
| 2.12 Browse For Folder .....   | 56   |
| 2.13 Update Driver Software 2 .....                                    | 57   |
| 2.14 Windows Security .....  | 57   |
| 2.15 Update Driver Software 3 .....                                    | 58   |
| 2.16 Folder Arduino 1.0.1 .....  | 58   |
| 2.17 Arduino 1.0.1 .....   | 59   |
| 3.2 แผนภาพ Flowchart ขั้นตอนการดำเนินงาน .....                         | 63   |
| 3.2 แผนภาพ Flowchart ขั้นตอนการดำเนินงาน (ต่อ) .....                   | 64   |
| 3.3 แสดง Flowchart รูปแบบการทำงานของวงจร.....                          | 65   |
| 4.1 การพิมพ์คำสั่งโค้ด Arduino ประกาศค่าของ Board Arduino .....        | 69   |
| 4.2 การพิมพ์คำสั่งโค้ด Arduino รับค่า เซนเซอร์ Ultrasonic .....        | 70   |
| 4.3 การพิมพ์คำสั่งโค้ด Arduino คำนวณระยะทาง Ultrasonic .....           | 70   |

## บทที่ 1

### บทนำ

#### 1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันเทคโนโลยีสารสนเทศเข้ามามีบทบาทในการใช้ชีวิตประจำวันของเรามากขึ้นทั้งด้านติดต่อสื่อสาร การเดินทาง การใช้เทคโนโลยีเพื่อค้นหาสิ่งต่าง ๆ สิ่งเหล่านี้ทำให้เกิดความสะดวกสบายต่อผู้คนต่างจากในสมัยก่อนการจะใช้อินเทอร์เน็ตเข้าถึงได้ยากมากและการสื่อสารผ่านอินเทอร์เน็ตในสมัยก่อนยังไม่ค่อยเป็นที่รู้จัก แต่ในปัจจุบันเทคโนโลยีได้มีการพัฒนาเทคโนโลยีพัฒนาอย่างรวดเร็วทำให้สิ่งต่าง ๆ รอบตัวถูกพัฒนาให้เข้าชุมชนและสังคมในปัจจุบันเทคโนโลยีเข้ามาช่วยในด้านการศึกษาในการค้นหาข้อมูลและอินเทอร์เน็ตก็จะมีบทบาทต่อผู้ใช้งานทำให้การค้นหาข้อมูลสามารถหาได้ง่ายและสะดวกรวดเร็วมากยิ่งขึ้นและบางอย่างสามารถนำมาช่วยให้ผู้พิการในด้านต่าง ๆ

ซึ่งงานนี้สร้างขึ้นเพื่อช่วยเหลือผู้พิการทางสายตาเพื่อช่วยลดการเกิดอุบัติเหตุรอบข้างและลดอุปสรรคในการเดินทางเพื่อช่วยให้แบ่งเบาภาระการใช้ไม้เท้าสำหรับผู้พิการทางสายตา เพราะชิ้นงานชิ้นนี้จะใช้เป็นหมวกแทนและจะมีเซ็นเซอร์ Ultrasonic ติดอยู่ซึ่งจะใช้ควบคู่กับ Board Arduino เขียนชุดคำสั่งควบคู่โดยใช้ภาษา C++ ตัวเซ็นเซอร์จะตรวจจับสิ่งกีดขวาง ภายในระยะ 2 เมตร ในทางแนวตรงภายในหมวกจะมีสัญญาณเตือนตัว Arduino ใช้ไฟเลี้ยง 9.0 V และมี Switch ใช้เปิด/ปิด Arduino และ Switch จะอยู่ด้านหลังของหมวก

ดังนั้นโครงการหมวกเซ็นเซอร์ตรวจจับสิ่งกีดขวางสำหรับผู้พิการทางสายตาเป็นโครงการที่มีประโยชน์ต่อผู้พิการทางสายตาและเป็นเสมือนอุปกรณ์ที่ใช้เตือนผู้บกพร่องทางร่างกาย ทางคณะผู้จัดทำโครงการได้มุ่งเน้นชิ้นงานไปทางการสร้างความปลอดภัยและลดอุบัติเหตุที่ผู้ใช้งานและผู้พิการทางสายตาไม่สามารถมองเห็นได้ โครงการหมวกเซ็นเซอร์ตรวจจับสิ่งกีดขวางสำหรับผู้พิการทางสายตาเป็นการเสริมสร้างทักษะทางด้านต่างๆ เช่น การทำอุปกรณ์ ด้านการเขียนชุดคำสั่ง การวิเคราะห์งาน การแก้ไข ปัญหา และการทำงานเป็นกลุ่ม ทางคณะผู้จัดทำโครงการหวังว่าโครงการที่จัดทำจะเป็นประโยชน์ต่อสังคมมากยิ่งขึ้น

#### 1.2 วัตถุประสงค์ของโครงการ

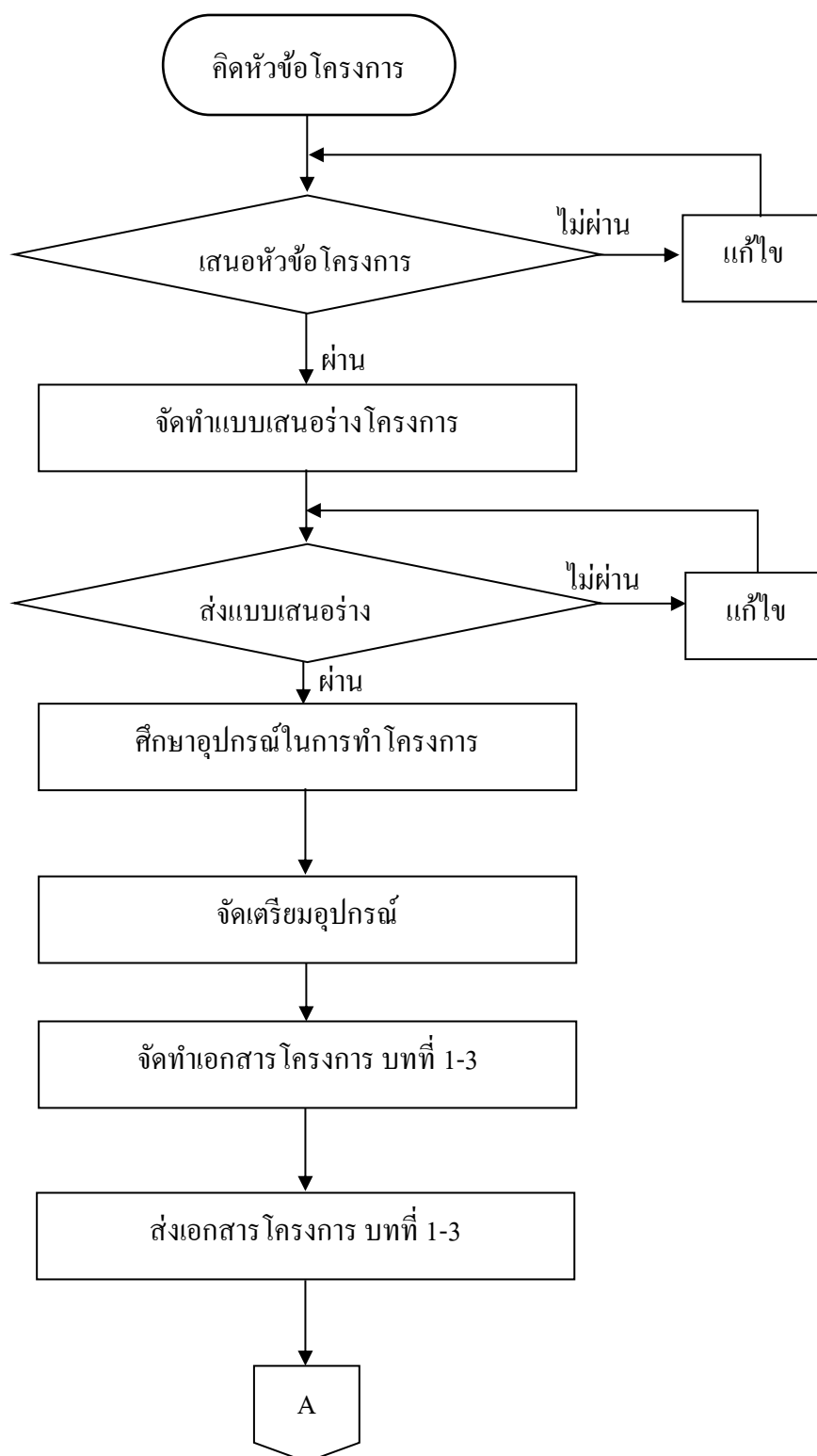
- 1.2.1 เพื่อให้ผู้พิการทางสายตาคลาดปัญหาและอุปสรรคสำหรับการเดินทาง
- 1.2.2 เพื่อให้ผู้จัดทำโครงการนำความรู้จากการศึกษาในหลักสูตรของประกาศนียบัตรวิชาชีพ มาประยุกต์ใช้ในการทำโครงการ
- 1.2.3 เพื่อลดการอุบัติเหตุแก่ผู้พิการทางสายตา

### 1.3 ขอบเขตของโครงการ

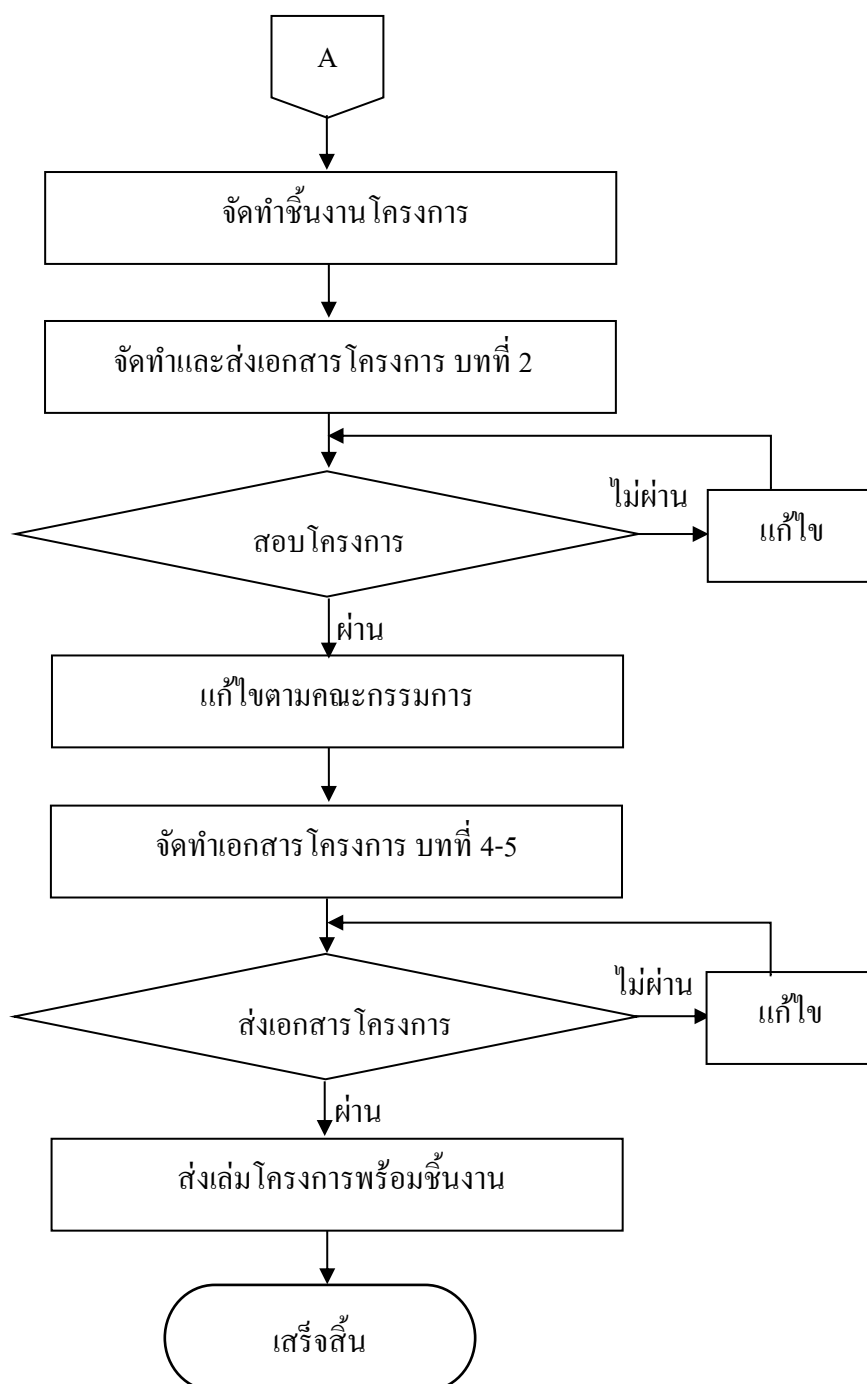
- 1.3.1 ใช้โปรแกรมArduinoในการเขียนชุดคำสั่ง
- 1.3.2 ใช้ Code คำสั่ง โปรแกรม Arduino C++ version1.8.9
- 1.3.3 ใช้ภาษา C++ ในการเขียนโปรแกรม Arduino

### 1.4 ขั้นตอนการดำเนินงาน

- 1.4.1 คิดหัวข้อโครงการ
- 1.4.2 จัดทำแบบเสนอร่างโครงการ
- 1.4.3 ส่งแบบเสนอโครงการ
- 1.4.4 ศึกษาอุปกรณ์ในการทำโครงการ
- 1.4.5 จัดเตรียมอุปกรณ์
- 1.4.6 จัดทำเอกสารโครงการ บทที่ 1-3
- 1.4.7 ส่งเอกสารโครงการ บทที่ 1-3
- 1.4.8 จัดทำชิ้นงานโครงการ
- 1.4.9 จัดทำเอกสารโครงการ บทที่ 2
- 1.4.10 ส่งเอกสารโครงการ บทที่ 2
- 1.4.11 ทดสอบการทำงานของชิ้นงาน
- 1.4.12 สอบโครงการ
- 1.4.13 แก้ไขตามคณะกรรมการ
- 1.4.14 ส่งชิ้นงานให้กรรมตรวจสอบ
- 1.4.15 จัดทำเอกสารโครงการ บทที่ 4-5
- 1.4.16 ส่งเอกสารโครงการ บทที่ 4-5
- 1.4.17 ส่งเล่มโครงการพร้อมชิ้นงาน



รูปที่ 1.1 แผนภาพ Flowchart ขั้นตอนการดำเนินงาน



รูปที่ 1.1 แผนภาพ Flowchart ขั้นตอนการดำเนินงาน (ต่อ)



### 1.6 ระยะเวลาการทำโครงการ

ตารางการดำเนินงานโครงการนี้ใช้ระยะเวลาการพัฒนา ตั้งแต่เดือน มิถุนายน พ.ศ. 2562 ถึงเดือนกุมภาพันธ์ พ.ศ. 2563 ดังตารางที่ 1.1

ตารางที่ 1.1 ตารางแผนการดำเนินงาน

| ลำดับ | ขั้นตอนการดำเนินงาน          | ระยะเวลาการดำเนินงาน |      |      |      |      |      |      |              |      |
|-------|------------------------------|----------------------|------|------|------|------|------|------|--------------|------|
|       |                              | ปี พ.ศ. 2562         |      |      |      |      |      |      | ปี พ.ศ. 2563 |      |
|       |                              | มิ.ย.                | ก.ค. | ส.ค. | ก.ย. | ต.ค. | พ.ย. | ธ.ค. | ม.ค.         | ก.พ. |
| 1     | คิดหัวข้อโครงการ             | ↔                    |      |      |      |      |      |      |              |      |
| 2     | นำเสนอโครงการ                | ↔                    |      |      |      |      |      |      |              |      |
| 3     | ส่งแบบเสนอโครงการ            | ↔                    |      |      |      |      |      |      |              |      |
| 4     | ศึกษาอุปกรณ์ในการทำโครงการ   | ↔                    |      |      |      |      |      |      |              |      |
| 5     | จัดเตรียมอุปกรณ์             | ↔                    | ↔    |      |      |      |      |      |              |      |
| 6     | จัดทำเอกสารโครงการ บทที่ 1-3 |                      | ↔    | ↔    |      |      |      |      |              |      |
| 7     | ส่งเอกสารโครงการ บทที่ 1-3   |                      |      |      | ↔    |      |      |      |              |      |
| 8     | จัดทำชิ้นงานโครงการ          |                      |      |      | ↔    | ↔    |      |      |              |      |

ตารางที่ 1.1 ตารางแผนการดำเนินงาน (ต่อ)

| ลำดับ | ขั้นตอนการดำเนินงาน                          | ระยะเวลาการดำเนินงาน |      |      |      |      |      |      |              |      |
|-------|--|----------------------|------|------|------|------|------|------|--------------|------|
|       |  | ปี พ.ศ. 2562         |      |      |      |      |      |      | ปี พ.ศ. 2563 |      |
|       |  | มิ.ย.                | ก.ค. | ส.ค. | ก.ย. | ต.ค. | พ.ย. | ธ.ค. | ม.ค.         | ก.พ. |
| 9     | จัดทำเอกสาร<br>โครงการ<br>บทที่ 2            |                      |      | ↔    |      |      |      |      |              |      |
| 10    | ส่งเอกสารโครงการ<br>บทที่ 2                  |                      |      |      | ↔    |      |      |      |              |      |
| 11    | ทดสอบการทำงาน<br>ของชิ้นงาน                  |                      |      |      |      | ↔    |      |      |              |      |
| 12    | สอบโครงการ                                   |                      |      |      |      |      | ↔    |      |              |      |
| 13    | แก้ไขตามที่คณะกรรมการ                        |                      |      |      |      |      | ↔    |      |              |      |
| 14    | ส่งชิ้นงานให้<br>กรรมการตรวจ<br>จสอบ         |                      |      |      |      |      | ↔    |      |              |      |
| 15    | จัดทำเอกสาร<br>โครงการ<br>บทที่ 4-5          |                      |      |      |      | ↔    |      |      |              |      |
| 16    | ส่งเอกสารโครงการ<br>บทที่ 4-5                |                      |      |      |      |      |      | ↔    |              |      |
| 17    | ส่งเล่มโครงการ<br>พร้อมชิ้นงาน<br>ที่สมบูรณ์ |                      |      |      |      |      |      |      | ↔            |      |

1.6 ประโยชน์ที่คาดว่าจะได้รับ

- 1.6.1 ผู้พิการทางสายตาลดปัญหาและอุปสรรคสำหรับการเดินทาง
- 1.6.2 ให้ผู้จัดทำโครงการนำความรู้จากการศึกษาในหลักสูตรของประกาศนียบัตรวิชาชีพมาประยุกต์ใช้ในโครงการ
- 1.6.3 ลดเกิดการอุบัติเหตุแก่ผู้พิการทางสายตา

#### 1.7 งบประมาณที่ใช้ในการทำโครงการ

|                          |                            |            |
|--------------------------|----------------------------|------------|
| 1.7.1 ค่าหมึกพิมพ์เอกสาร | 1,200                      | บาท        |
| 1.7.2 ค่ากระดาษ A4       | 100                        | บาท        |
| 1.7.3 ค่าแผ่นDVD         | 50                         | บาท        |
| 1.7.4 ค่าเล่มโครงการ     | <u>200</u>                 | บาท        |
| <b>รวม</b>               | <b><u><u>1,550</u></u></b> | <b>บาท</b> |

## บทที่ 2

### เอกสารและทฤษฎีที่เกี่ยวข้อง

ในการศึกษาผลงานที่เกี่ยวข้องกับการสร้างหมวกเซนเซอร์ตรวจจับสิ่งกีดขวางสำหรับผู้พิการทางสายตา ซึ่งนับว่าเป็นองค์ประกอบหนึ่งที่มีความสำคัญมาก จะทำให้โครงการครบถ้วนสมบูรณ์มากยิ่งขึ้นองค์ประกอบหนึ่งที่มีความสำคัญมาก จะทำให้โครงการมีความครบถ้วนสมบูรณ์

คณะผู้จัดทำได้แบ่งเอกสารและทฤษฎีที่เกี่ยวข้องออกเป็นหัวข้อ ดังนี้

- 2.1 ความต้องการของระบบที่เหมาะสม
- 2.2 สอนใช้งาน Arduino ควบคุม ultrasonic วัดระยะทาง
- 2.3 ภาษาซี (C Programing Language)
- 2.4 การใช้ if else
- 2.5 การคำนวณระยะทาง
- 2.6 วิธีการติดตั้ง Driver Arduino

#### 2.1 ความต้องการของระบบที่เหมาะสม

- 2.1.1 เครื่องคอมพิวเตอร์ควรติดตั้งโปรแกรม Arduino IDE
- 2.1.2 เครื่องคอมพิวเตอร์ควรมีความจุ (Ram) 512 MB ขึ้นไป
- 2.1.3 เครื่องคอมพิวเตอร์ควรมีพื้นที่ว่างของอุปกรณ์บรรจุข้อมูล (Hard Disk) 100 GB ขึ้นไปเพื่อรองรับและบันทึกข้อมูลต่างๆ

#### 2.2 สอนใช้งาน Arduino ควบคุม ultrasonic วัดระยะทาง

- 2.2.1 สอนใช้งาน Arduino ควบคุม ultrasonic วัดระยะทาง

บทความนี้จะสอนใช้งาน NodeMCU ESP8266 เชื่อมต่อ Sensor วัดระยะทาง Ultrasonic Module HC-SR04 ตัว Sensor จะส่งคลื่นออกไปแล้วจับเวลาที่คลื่นกระทบกับวัตถุสะท้อนกลับมาเข้าตัวรับ ทำให้สามารถวัดระยะทางระหว่างวัตถุถึงตัวเซ็นเซอร์วัดระยะทาง Ultrasonic Module HC-SR04 ได้

```

const int pingPin = 13;
int inPin = 12;

void setup() {
  Serial.begin(9600);
}

void loop()
{
  long duration, cm;

  pinMode(pingPin, OUTPUT);

  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(5);
  digitalWrite(pingPin, LOW);
  pinMode(inPin, INPUT);
  duration = pulseIn(inPin, HIGH);

  cm = microsecondsToCentimeters(duration);

```

รูปที่ 2.1 ตัวอย่างชุดคำสั่ง ultrasonic sensor ที่มา([https://cu.lnwfile.com raw/x7/nv/pj.jpg](https://cu.lnwfile.com/raw/x7/nv/pj.jpg))

## 2.2.2 ตัวอย่างการใช้งาน เซนเซอร์วัดระยะทาง HC-SR04 Ultrasonic Module HC-SR04 -

> Arduino

Vcc -> 5v

Gnd -> Gnd

Trig -> 13

Echo -> 12

## 2.3 ภาษาซี (C Programming Language)

### 2.3.1 ภาษาซี

ภาษาซี (C) เป็นภาษาโปรแกรมสำหรับวัตถุประสงค์ทั่วไป เริ่มพัฒนาขึ้นระหว่าง พ.ศ. 2512–2516 (ค.ศ. 1969–1973) โดย เดนนิส ริตชี (Dennis Ritchie) ที่ เอลีแอนด์ทีเบลล์แล็บส์ (AT&T Bell Labs) ภาษาซีเป็นภาษาที่มีความยืดหยุ่นในการเขียนโปรแกรมและมีเครื่องมืออำนวยความสะดวกสำหรับการเขียนโปรแกรมเชิงโครงสร้างและอนุญาตให้มีขอบข่ายตัวแปร (scope) แดะการเรียกซ้ำ (recursion) ในขณะที่ระบบชนิดตัวแปรพลวัตก็ช่วยป้องกันการดำเนินการที่ไม่ตั้งใจหลายอย่าง เหมือนกับภาษาโปรแกรมเชิงคำสั่งส่วนใหญ่ในแบบแผนของภาษาอัลกอล การออกแบบของภาษาซีมีคอนสตรัคต์ (construct) ที่โยงกับชุดคำสั่งเครื่องทั่วไปได้อย่างพอเพียง

จึงทำให้ยังมีการใช้ในโปรแกรมประยุกต์ซึ่งแต่ก่อนลงรหัสเป็นภาษาแอสเซมบลีคือซอฟต์แวร์ระบบอันโดเด่นอย่างระบบปฏิบัติการคอมพิวเตอร์ ยูนิกซ์

ภาษาซีเป็นภาษาโปรแกรมหนึ่งที่ใช้กันอย่างแพร่หลายมากที่สุดตลอดกาล และตัวแปลโปรแกรมของภาษาซีมีให้ใช้งานได้สำหรับสถาปัตยกรรมคอมพิวเตอร์และระบบปฏิบัติการต่าง ๆ เป็นส่วนมาก

ภาษาหลายภาษาในยุคหลังได้หยิบยืมภาษาซีไปใช้ทั้งทางตรงและทางอ้อม ตัวอย่างเช่น ภาษาดี ภาษาโก ภาษาอาร์ส ภาษาจาวา จาวาสคริปต์ ภาษาลิมโบ ภาษาแอลพีซี ภาษาซีชาร์ป ภาษาเพิร์ล ภาษาพีเอชพี ภาษาไพทอน ภาษาเวริล็อก (ภาษาพรรณนาฮาร์ดแวร์) และซีเชลล์ของยูนิกซ์ ภาษาเหล่านี้ได้ดึงโครงสร้างการควบคุมและคุณลักษณะพื้นฐานอื่น ๆ มาจากภาษาซี ส่วนใหญ่มีวากยสัมพันธ์คล้ายคลึงกับภาษาซีเป็นอย่างมากโดยรวม ( ยกเว้น ภาษาไพทอน ที่ต่างออกไปอย่างสิ้นเชิง ) และตั้งใจที่จะผสานนิพจน์และข้อความสั่งที่จำแนกได้ของวากยสัมพันธ์ของภาษาซี ด้วยระบบชนิดตัวแปร ตัวอย่างแบบข้อมูลและอรรถศาสตร์ที่อาจแตกต่างกันโดยมูลฐาน ภาษาซีพลัสพลัสและภาษาอ็อบเจกทีฟ-ซี เดิมเกิดขึ้นในฐานะตัวแปลโปรแกรมที่สร้างรหัสภาษาซี ปัจจุบันภาษาซีพลัสพลัสแทบจะเป็นเซตใหญ่ของภาษาซี ในขณะที่ภาษาอ็อบเจกทีฟ-ซีก็เป็นเซตใหญ่อันเคร่งครัดของภาษาซี

ก่อนที่จะมีมาตรฐานภาษาซีอย่างเป็นทางการ ผู้ใช้และผู้พัฒนาต่างก็เชื่อถือในข้อกำหนดอย่างไม่เป็นทางการในหนังสือที่เขียนโดยเคนนิส ริดชี และไบรอัน เคอร์นิกัน ( Brian Kernighan) ภาษาซีรุ่นนั้นจึงเรียกกันโดยทั่วไปว่า ภาษาเคแอนด์อาร์ซี (K&R C) ต่อมา พ.ศ. 2532 สถาบันมาตรฐานแห่งชาติของสหรัฐอเมริกา (ANSI) ได้ตีพิมพ์มาตรฐานสำหรับภาษาซีขึ้นมา เรียกกันว่า ภาษาแอนซีซี ( ANSI C) หรือ ภาษาซี 89 ( C89) ในปีถัดมา องค์การระหว่างประเทศว่าด้วยการมาตรฐาน ( ISO) ได้อนุมัติให้ข้อกำหนดเดียวกันนี้เป็นมาตรฐานสากล เรียกกันว่า ภาษาซี 90 ( C90) ในเวลาต่อมาอีกองค์การฯ ก็ได้เผยแพร่ส่วนขยายมาตรฐานเพื่อรองรับสากลวิวัฒน์ เมื่อ พ.ศ. 2538 และมาตรฐานที่ตรวจชำระใหม่เมื่อ พ.ศ. 2542 เรียกกันว่า ภาษาซี 99 ( C99) มาตรฐานรุ่นปัจจุบันก็ได้รับอนุมัติเมื่อเดือนธันวาคม พ.ศ. 2554 เรียกกันว่า ภาษาซี 12 ( C12)

### 2.3.2 การออกแบบ

ภาษาซีเป็นภาษาที่ใช้ในการมีปฏิสัมพันธ์เช่น เชิงคำสั่ง (หรือเชิงกระบวนการ) ถูกออกแบบขึ้นเพื่อใช้แปลด้วยตัวแปลโปรแกรมแบบการเชื่อมโยงที่ตรงไปตรงมา สามารถเข้าถึงหน่วยความจำในระดับล่าง เพื่อสร้างภาษาที่จับคู่อย่างมีประสิทธิภาพกับชุดคำสั่งเครื่อง และแทบไม่ต้องการสนับสนุนใดๆ ขณะทำงาน ภาษาซีจึงเป็นประโยชน์สำหรับหลายโปรแกรมที่ก่อนหน้านี้เคยเขียนในภาษาแอสเซมบลีมาก่อน

หากคำนึงถึงความสามารถในระดับล่าง ภาษานี้ถูกออกแบบขึ้นเพื่อส่งเสริมการเขียนโปรแกรมที่ขึ้นอยู่กับเครื่องใดเครื่องหนึ่ง (machine-independent) โปรแกรมภาษาซีที่เขียนขึ้นตามมาตรฐานและเคลื่อนย้ายได้ สามารถแปลได้บนแพลตฟอร์มคอมพิวเตอร์และระบบปฏิบัติการต่างๆ อย่างกว้างขวาง โดยแก้ไขรหัสต้นฉบับเพียงเล็กน้อยหรือไม่ต้องแก้ไขเลย ภาษานี้สามารถใช้ได้บนแพลตฟอร์มได้หลากหลายตั้งแต่ไมโครคอนโทรลเลอร์ฝังตัวไปจนถึง[[แอสรีเคน]]-

### 2.3.3 ลักษณะเฉพาะ

ภาษาซีมีสิ่งอำนวยความสะดวกสำหรับการเขียนโปรแกรมเชิงโครงสร้าง และสามารถกำหนดและเรียกซ้ำ เช่นเดียวกับภาษาโปรแกรมเชิงคำสั่งส่วนใหญ่ในสายตระกูลภาษาอัลกอล ในขณะที่ระบบชนิดตัวแปรแบบอพลวัตช่วยป้องกันการดำเนินการที่ไม่ได้ตั้งใจ รหัสที่ทำงานได้ทั้งหมดในภาษาซีถูกบรรจุอยู่ในฟังก์ชัน พารามิเตอร์ของฟังก์ชันส่งผ่านด้วยค่าของตัวแปรและโครงสร้าง ส่วนการส่งผ่านด้วยการอ้างอิงจะถูกจำลองขึ้นโดยการส่งผ่านค่าตัวชี้ ชนิดข้อมูลรวมแบบแตกต่าง (Struct) ช่วยให้สมาชิกข้อมูลที่เกี่ยวข้องกันสามารถรวมกันและจัดการได้ในหน่วยเดียว รหัสต้นฉบับของภาษาซีเป็นรูปแบบอิสระ ซึ่งใช้ฉลาก (;) เป็นตัวจบคำสั่ง (มิใช่ตัวแบ่ง)

ภาษาซียังมีลักษณะเฉพาะต่อไปนี้เพิ่มเติม

#### 2.3.3.1 ตัวแปรอาจถูกซ่อนในบล็อกซ้อนใน

#### 2.3.3.2 ชนิดตัวแปรไม่เคร่งครัด เช่นข้อมูลตัวอักษรสามารถใช้เป็นจำนวนเต็ม

2. 3. 3

เข้าถึงหน่วยความจำคอมพิวเตอร์ในระดับต่ำโดยแปลงที่อยู่บนเครื่องด้วยชนิดตัวแปรตัวชี้ (pointer)

2. 3. 4

ฟังก์ชันและตัวชี้ข้อมูลรองรับการทำงานในภาวะหลายรูปแบบ (polymorphism)

2.3.3.5 การกำหนดดัชนีแถวลำดับสามารถทำได้ด้วยวิธีรอง  
คือนิยามในพจน์ของเลขคณิตของตัวชี้

2.3.3.6 ตัวประมวลผลก่อนสำหรับการนิยามแมโคร การรวมไฟล์รหัสต้นฉบับ  
และการแปลโปรแกรมแบบมีเงื่อนไข

2. 3. 3. 7  
ความสามารถที่ซับซ้อนเช่น ไอ/โอ การจัดการสายอักขระ และฟังก์ชันทาง คณิตศาสตร์  
รวมอยู่ในไลบรารี

2.3.3.8 คำหลักที่สงวนไว้มีจำนวนค่อนข้างน้อย

2.3.3.9 ตัวดำเนินการแบบประสมจำนวนมาก เช่น +=, -=, \*=, ++ ฯลฯ  
โครงสร้างการเขียน คล้ายภาษาพีมากกว่าภาษาอัลกอล ตัวอย่างเช่น

2. 3. 3. 10 ใช้วงเล็บปีกกาแทนที่จะเป็นในภาษาอัลกอล  
60 หรือวงเล็บโค้งในภาษาอัลกอล 68

2. 3. 3. 11 เท่ากับ = ใช้สำหรับกำหนดค่า (คัดลอกข้อมูล)  
เหมือนภาษาฟอร์แทรน แทนที่จะเป็น := ในภาษาอัลกอล

2.3.3.12 เท่ากับสองตัว == ใช้สำหรับเปรียบเทียบความเท่ากัน แทนที่จะเป็น .EQ  
ในภาษาฟอร์แทรนหรือ = ในภาษาเบสิกและภาษาอัลกอล

2.3.3.13 ตรรกะ "และ" กับ "หรือ" แทนด้วย && กับ || ตามลำดับ  
แทนที่จะเป็นตัวดำเนินการ  $\wedge$  กับ  $\vee$  ในภาษาอัลกอล  
แต่ตัวดำเนินการดังกล่าวจะไม่ประเมินค่าตัวถูกดำเนินการทางขวา  
ถ้าหากผลลัพธ์จากทางซ้ายสามารถพิจารณาได้แล้ว  
เหตุการณ์เช่นนี้เรียกว่าการประเมินค่าแบบลัดวงจร (short-circuit evaluation)  
และตัวดำเนินการดังกล่าวก็มีความหมายต่างจากตัวดำเนินการระดับบิต & กับ I

2.3.4 คุณลักษณะที่ขาดไป

ธรรมชาติของภาษาในระดับต่ำช่วยให้โปรแกรมเมอร์ควบคุมสิ่งที่คอมพิวเตอร์กระทำได้อ  
ย่ ำ ง ใ ก ล้ ชี ด  
ในขณะที่อนุญาตให้มีการปรับแต่งพิเศษและการทำให้เหมาะสมที่สุดสำหรับแพลตฟอร์มหนึ่งใดโ  
ค ย เ ณ พ ำ ะ  
สิ่งนี้ทำให้รหัสสามารถทำงานได้อย่างมีประสิทธิภาพบนฮาร์ดแวร์ที่มีทรัพยากรจำกัดมาก ๆ  
ได้เช่นระบบฝังตัว

2.3.4.1 ภาษาซีไม่มีคุณลักษณะบางอย่างที่มีในภาษาอื่น อาทิ

2.3.4.2 ไม่มีการนิยามฟังก์ชันซ้อนใน



2.3.4.3 ไม่มีการกำหนดค่าแถวลำดับหรือสายอักขระโดยตรง  
(การคัดลอกข้อมูลจะกระทำผ่านฟังก์ชันมาตรฐานกำหนดค่าวัตถุที่มีชนิดเป็น Struct หรือ union)

2.3.4.4 ไม่มีการเก็บข้อมูลขยะโดยอัตโนมัติ

2.3.4.5 ไม่มีข้อกำหนดเพื่อการตรวจสอบขอบเขตของแถวลำดับ

2.3.4.6 ไม่มีการดำเนินการสำหรับแถวลำดับทั้งหมดในระดับตัวภาษา

2.3.4.7 ไม่มีวากยสัมพันธ์สำหรับช่วงค่า (range) เช่น A..B ที่ใช้ในบางภาษา

2. 3. 4. 8 ก่ อ น ถึ ง ภ า ษ า ษี 99

ไม่มีการแบ่งแยกชนิดข้อมูลแบบบูล (ค่าศูนย์หรือไม่ศูนย์ถูกนำมาใช้แทน)

2.3.4.9 ไม่มีส่วนปิดคลุมแบบรูปนัย (closure) หรือฟังก์ชันในรูปแบบพารามิเตอร์  
(มีเพียงตัวชี้ของฟังก์ชันและตัวแปร)

2. 3. 4. 10

ไม่มีตัวสร้างและโครูทีน การควบคุมกระแสการทำงานภายในเทรีดมีเพียงการเรียกใช้ฟังก์ชันซ้อน  
ลงไป เว้นแต่การใช้ฟังก์ชัน long หรือ setcontext จากไลบรารี

2.3.4.11 ไม่มีการจัดการทำสิ่งผิดปกติ (exception handling)  
ฟังก์ชันไลบรารีมาตรฐานจะแสดงเงื่อนไขข้อผิดพลาดด้วยตัวแปรส่วนกลาง errno  
และ/หรือค่ากลับคืนพิเศษ และฟังก์ชันไลบรารีได้เตรียม goto แบบไม่ใช่เฉพาะที่ไว้ด้วย

2.3.4.12 การเขียนโปรแกรมเชิงมอดูลรองรับแค่ระดับพื้นฐานเท่านั้น

2.3.4.13 การโอเวอร์โหลดฟังก์ชันหรือตัวดำเนินการไม่รองรับภาวะหลายรูปแบบ  
ขณะแปลโปรแกรม

2.3.4.14 การเขียนโปรแกรมเชิงวัตถุรองรับในระดับที่จำกัดมาก  
โดยพิจารณาจากภาวะหลายรูปแบบกับการรับทอด (inheritance)

2.3.4.15 การซ่อนสารสนเทศ (encapsulation) รองรับในระดับที่จำกัด

2. 3. 4. 16

ไม่รองรับโดยพื้นฐานกับการทำงานแบบมัลติเทรีดและเครือข่ายคอมพิวเตอร์

2. 3. 4. 17

ไม่มีไลบรารีมาตรฐานสำหรับคอมพิวเตอร์กราฟิกส์และความจำเป็นหลายอย่างในการเขียนโปรแกรมประยุกต์

2.3.5 สิ่งที่ไม่เกิดขึ้น

การดำเนินการหลายอย่างในภาษาซีมีพฤติกรรมไม่นิยามซึ่งไม่ถูกกำหนดว่าต้องตรวจสอบ  
ขณะแปลโปรแกรม ในกรณีของภาษาซี "พฤติกรรมไม่นิยาม"  
หมายถึงพฤติกรรมเฉพาะอย่างที่เกิดขึ้นโดยมาตรฐานมิได้ระบุไว้

และ สิ่ง ที่ จะ เกิด ขึ้น ก็ ไม่ มี ใน เอก สาร การ ใช้งาน ของ ภาษา ซี  
หนึ่งในชุดคำสั่งที่มีชื่อเสียงและน่าขบขันจากกลุ่มข่าว comp.std.c และ comp.lang.c นั้นทำให้โปรแกรมเกิดปัญหาที่เรียกว่า "ปิศาจที่ออกมาจากจมูกของคุณ" (demons to fly out of your nose) บางครั้งสิ่งที่เกิดขึ้นในทางปฏิบัติอันเป็นผลมาจากพฤติกรรมไม่นิยามทำให้เกิดจุดบกพร่องที่ยากต่อการตรวจสอบและอาจทำให้ข้อมูลในหน่วยความจำผิดแปลกไป  
ตัวแปลโปรแกรมบางชนิดช่วยสร้างการดำเนินงานที่ทำให้พฤติกรรมนั้นดีขึ้นและมีเหตุผลซึ่งแตกต่างจากการแปลโดยตัวแปลชนิดอื่นที่อาจดำเนินงานไม่เหมือนกัน  
สาเหตุที่พฤติกรรมบางอย่างยังคงไว้ว่าไม่นิยามก็เพื่อให้ตัวแปลโปรแกรมบนสถาปัตยกรรมชุดของคำสั่งจะรู้เองที่หลีกเลี่ยง  
สามารถสร้างรหัสที่ทำงานได้ในพฤติกรรมที่นิยามอย่างมีประสิทธิภาพมากขึ้น  
ซึ่งเชื่อว่าเป็นบทบาทหนึ่งที่สำคัญของภาษาซีในฐานะภาษาสำหรับสร้างระบบ  
ดังนั้นภาษาซีจึงส่งผลให้เกิดความรับผิดชอบของโปรแกรมเมอร์เพื่อหลีกเลี่ยงพฤติกรรมไม่นิยามโดยอาจใช้เครื่องมือต่าง ๆ เพื่อค้นหาส่วนของโปรแกรมว่าพฤติกรรมใดบ้างที่ไม่นิยาม  
ตัวอย่างของพฤติกรรมไม่นิยามเช่น

2.3.5.1 การเข้าถึงข้อมูลนอกขอบเขตของแฉวลำดับ

2.3.5.2 ข้อมูลล้น (overflow) ในตัวแปรจำนวนเต็มมีเครื่องหมาย

2.3.5.3 ฟังก์ชันที่กำหนดไว้ว่าต้องส่งค่ากลับ แต่ไม่มีคำสั่งส่งกลับ (return)

ในฟังก์ชันในขณะเดียวกันค่าส่งกลับก็ถูกใช้งานด้วย

2.3.5.4 การอ่านค่าตัวแปรโดยที่ยังไม่ได้กำหนดค่าเริ่มต้น

การดำเนินการเหล่านี้ทั้งหมดเป็นข้อผิดพลาดในการเขียนโปรแกรมซึ่งสามารถปรากฏในการใช้ภาษาโปรแกรมอื่น ๆ จำนวนมาก  
ภาษาซีจึงถูกวิพากษ์วิจารณ์เพราะมาตรฐานของมันสามารถชี้ให้เห็นถึงพฤติกรรมไม่นิยามในหลายกรณีได้อย่างชัดเจน รวมไปถึงพฤติกรรมบางอย่างที่อาจนิยามไว้อย่างดีแล้วและไม่มีการระบุกลไกการกระทำต่อข้อผิดพลาดขณะทำงานเลย

ตัวอย่างหนึ่งของพฤติกรรมไม่นิยามเช่นการเรียกใช้ fflush() บนกระแสข้อมูลป้อนเข้าซึ่งไม่จำเป็นว่าจะทำให้โปรแกรมทำงานผิดพลาดแต่ในบางกรณีที่ทำให้เกิดผลที่สอดคล้องกันได้นิยามไว้แล้วอย่างดีมีความหมายซึ่งใช้ประโยชน์ได้ (จากตัวอย่างนี้คือการสมมติให้ข้อมูลที่ป้อนเข้าถูกละทิ้งทั้งหมดจนถึงอักขระขึ้นบรรทัดใหม่ตัวถัดไป) เป็นส่วนขยายที่อนุญาตส่วนขยายที่ไม่เป็นมาตรฐานเช่นนี้เป็นข้อจำกัดความสามารถในการเคลื่อนย้ายของซอฟต์แวร์

### 2.3.6 ประวัติภาษาซี

#### การพัฒนาช่วงแรก

การเริ่มต้นพัฒนาภาษาซีเกิดขึ้นที่เบลล์แล็บส์ของเอทีแอนด์ทีระหว่าง พ.ศ. 2512–2516 แต่ตามข้อมูลของริตชี ช่วงเวลาที่เกิดความสร้างสรรค์มากที่สุดคือ พ.ศ. 2515 ภาษานี้ถูกตั้งชื่อว่า "ซี" เพราะคุณลักษณะต่าง ๆ ต่อยอดมาจากภาษาก่อนหน้าคือ "บี" ซึ่งมาจากข้อมูลของเคน ทอมป์สัน (Ken Thompson) กล่าวว่าภาษาบีเป็นรุ่นที่แยกตัวออกจากภาษาบีซีพีแอลอีกทอดหนึ่ง

จุดเริ่มต้นของภาษาซีผูกอยู่กับการพัฒนาระบบปฏิบัติการยูนิกซ์อย่างใกล้ชิด ซึ่งเดิมพัฒนาด้วยภาษาแอสเซมบลีบนหน่วยประมวลผลพีดีพี-7 โดยริตชีและทอมป์สัน โดยผสมผสานความคิดหลากหลายจากเพื่อนร่วมงานในตอนท้ายพวกเขาตัดสินใจที่จะย้ายระบบปฏิบัติการนั้นลงในพีดีพี-11 แต่ภาษาบีขาดความสามารถบางอย่างที่จะใช้คุณลักษณะอันได้เปรียบของพีดีพี-11 เช่นความสามารถในการระบุตำแหน่งที่อยู่เป็นไบต์ จึงทำให้เกิดการพัฒนาภาษาซีรุ่นแรกขึ้นมา

รุ่นดั้งเดิมของระบบยูนิกซ์บนพีดีพี-11ถูกพัฒนาขึ้นด้วยภาษาแอสเซมบลี เมื่อประมาณ พ.ศ. 2516 ภาษาซีเพิ่มชนิดข้อมูล Struct ทำให้ภาษาซีเพียงพออย่างมีประสิทธิภาพ ซึ่งเคอร์เนलयูนิกซ์ส่วนใหญ่ถูกเขียนด้วยภาษาซี นี่ก็เป็นเคอร์เนลหนึ่งของระบบปฏิบัติการที่พัฒนาด้วยภาษาอื่นนอกเหนือจากภาษาแอสเซมบลี (ระบบอื่นเช่นมัลติคส์เขียนด้วยภาษาพีแอล/วัน เอ็มซีพีสำหรับเบอร์โรส์ปี5000เขียนด้วยภาษาอัลกอล ในปี พ.ศ. 2504)

#### 2.3.7 ภาษาเคแอนด์อาร์ซี

เมื่อ พ.ศ. 2521 ไบรอัน เคอร์นิกัน (Brian Kernighan) และเดนนิส ริตชี ได้ตีพิมพ์หนังสือเล่มแรกชื่อ เดอะซีโปรแกรมมิงแลงกวิจ (The C Programming Language) ซึ่งเป็นที่รู้จักในกลุ่มโปรแกรมเมอร์ภาษาซีว่า "เคแอนด์อาร์" (K&R อักษรย่อของ ผู้แต่งทั้งสอง) หนังสือเล่มนี้ทำหน้าที่เป็นข้อกำหนดของภาษาอย่างไม่เป็นทางการมาหลายปี ภาษาซีรุ่นดังกล่าวจึงมักถูกอ้างถึงว่าเป็น ภาษาเคแอนด์อาร์ซี (K&R C) ส่วนหนังสือที่ปรับปรุงครั้งที่สองครอบคลุมมาตรฐานแอนซีซีที่มีขึ้นทีหลัง

ภาษาเคแอนด์อาร์ซีได้นำคุณลักษณะหลายประการเช่น

##### 2.3.7.1 ไบรารีไอ/โอมาตรฐาน

##### 2.3.7.2 ชนิดข้อมูล long int (จำนวนเต็มขนาดยาว)

### 2.3.7.3 ชนิดข้อมูล unsigned int(จำนวนเต็มไม่มีเครื่องหมาย)

2.3.7.4 ตัวดำเนินการกำหนดค่าแบบประสมในรูปแบบ =ตัวดำเนินการ -=) ถูกเปลี่ยนเป็น ตัวดำเนินการ= (เช่น -=) เพื่อลดปัญหาความกำกวมเชิงความหมาย อย่างเช่นกรณี I=10 ซึ่งจะถูกตีความว่า I=-10 แทนที่จะเป็นอย่างที่ตั้งใจคือ I = -10

แม้ว่าหลังจากการเผยแพร่มาตรฐานของภาษาซีเมื่อ พ.ศ. 2532 ภาษาเคแอลดีอาร์ซีถูกพิจารณาว่าเป็น "ส่วนร่วมต่ำสุด" อยู่เป็นเวลาหลายปี (ความสามารถในการแปลรหัสจำนวนหนึ่งเป็นคำสั่งซึ่งทำงานได้บนเครื่องใดก็ตามเป็นอย่างน้อย) ซึ่งโปรแกรมเมอร์ภาษาซีต้องจำกัดความสามารถของพวกเขาในกรณีที่ต้องการให้ระบบสามารถใช้ได้กับหลายเครื่องมากที่สุด เนื่องจากตัวแปลโปรแกรมเก่า ๆ ก็ยังคงมีการใช้งานอยู่ และการเขียนภาษาซีแบบเคแอลดีอาร์อย่างระมัดระวังสามารถเข้ากันได้กับภาษาซีมาตรฐานเป็นอย่างดี

ในภาษาซีรุ่นแรก ๆ เฉพาะฟังก์ชันที่คืนค่าไม่เป็นจำนวนเต็ม จำเป็นต้องประกาศไว้ก่อนการนิยามฟังก์ชันหากมีการเรียกใช้ อีกนัยหนึ่งคือ ฟังก์ชันที่ถูกเรียกใช้โดยไม่มีการประกาศมาก่อน ถือว่าฟังก์ชันนั้นจะคืนค่าเป็นจำนวนเต็มหากค่าของมันถูกใช้งาน ตัวอย่างเช่น

```
long int SomeFunction();
```

```
/* int OtherFunction(); */
```

```
/* int */ CallingFunction()
```

```
long int test1;
```

```
register /* int */ test2;
```

```
test1 = SomeFunction();
```

```
if (test1 > 0)
```

```
test2 = 0;
```

```
else
```

```
test2 = OtherFunction();
```

```
return test2;
```

จากตัวอย่างข้างต้น การประกาศ `int` ที่ถูกคัดลอก สามารถละเว้นได้ในภาษาเคแอลอาร์ซี แต่ `long int` จำเป็นต้องประกาศ

การประกาศฟังก์ชันของภาษาเคแอลอาร์ซีไม่มีการระบุข้อมูลเกี่ยวกับอาร์กิวเมนต์ที่ใช้ ดังนั้นจึงไม่มีการตรวจชนิดข้อมูลพารามิเตอร์ของฟังก์ชัน แม้ว่าตัวแปลโปรแกรมบางตัวจะแสดงข้อความเตือน ถ้าฟังก์ชันถูกเรียกใช้ภายในโดยมีจำนวนอาร์กิวเมนต์ที่ผิด หรือถ้าฟังก์ชันถูกเรียกใช้หลายครั้งจากภายนอกโดยมีชนิดข้อมูลของอาร์กิวเมนต์ต่างกัน เกลือจึงมีอภายนอกอาทิลินต์ (lint) ของยูนิกซ์ถูกพัฒนาขึ้นเพื่อให้สามารถตรวจสอบความคงเส้นคงวาของฟังก์ชันที่ใช้งานข้ามไฟล์หัตถ์ฉบับหลายไฟล์

หลายปีถัดจากการเผยแพร่ภาษาเคแอลอาร์ซีคุณลักษณะที่ไม่เป็นทางการหลายอย่างก็ถูกเพิ่มเข้ามาในภาษา ซึ่งรองรับโดยตัวแปลโปรแกรมจากเอทีแอนด์ทีและผู้ผลิตรายอื่น คุณลักษณะที่เพิ่มเหล่านี้เช่น

#### 2.3.7.1 ฟังก์ชัน void

#### 2.3.7.2 ฟังก์ชันที่คืนค่าเป็นชนิดข้อมูล `struct` หรือ `union` (แทนที่จะเป็นตัวชี้)

#### 2.3.7.3 การกำหนดค่าให้กับชนิดข้อมูล `struct`

#### 2.3.7.4 ชนิดข้อมูลแฉกนับ (enumerated type)

ส่วนขยายที่เพิ่มขึ้นอย่างมากและการขาดข้อตกลงในเรื่องไอบรรีมาตรฐาน อีกทั้งความนิยมในภาษาและข้อเท็จจริงที่ว่าไม่เพียงแต่ตัวแปลโปรแกรมยูนิกซ์เท่านั้นที่พัฒนาขึ้นตามข้อกำหนดของเคแอลอาร์ซี ทั้งหมดนำไปสู่ความสำคัญของการทำให้เป็นมาตรฐาน

### 2.3.8 การใช้งานระบบภาษาซี

การเขียนโปรแกรมระบบเป็นการใช้งานหลักของภาษาซี ซึ่งรวมไปถึงการพัฒนาระบบปฏิบัติการและโปรแกรมประยุกต์ระบบฝังตัว เนื่องจากลักษณะเฉพาะอันเป็นที่ต้องการถูกรวมเข้าไว้ด้วยกัน อย่างเช่น ความสามารถในการเคลื่อนย้ายได้กับประสิทธิภาพของรหัสต้นฉบับ ความสามารถในการเข้าถึงที่อยู่ของฮาร์ดแวร์ที่ระบุ ความสามารถเรื่อง `type punning` เพื่อให้เข้ากับความต้องการการเข้าถึงข้อมูลที่กำหนดไว้จากภายนอก และความต้องการทรัพยากรระบบขณะทำงานต่ำ ภาษาซีสามารถใช้เขียนโปรแกรมเว็บไซต์โดยใช้ซีจีไอเป็น "เกตเวย์" เพื่อแลกเปลี่ยนสารสนเทศระหว่างเว็บแอปพลิเคชัน เซิร์ฟเวอร์ และเบราว์เซอร์

ปัจจัยบางอย่างที่ทำให้เลือกภาษาซีแทนที่จะเป็นภาษาอินเทอร์พรีเตอร์ คือความเร็ว เสถียรภาพ และความอ่อนไหวต่อการเปลี่ยนแปลงในสภาพแวดล้อมของการดำเนินงาน เนื่องจากเป็นธรรมชาติของภาษาคอมไพเลอร์

ผลจากการยอมรับในระดับกว้างขวางและประสิทธิภาพของภาษาซีทำให้ตัวแปลโปรแกรม ตัวแปลคำสั่ง ไบบารีต่าง ๆ ของภาษาอื่น มักพัฒนาขึ้นด้วยภาษาซี ตัวอย่างเช่น ตัวแปลโปรแกรมภาษาไอเฟลหลายโปรแกรมส่งข้อมูลออกเป็นรหัสภาษาซีเป็นภาษากลางเพื่อส่งต่อไปให้ตัวแปลโปรแกรมภาษาซีต่อไป การพัฒนาสายหลักของภาษาไพทอน ภาษาเพิร์ล 5 และภาษาพีเอชพี ทั้งหมดถูกเขียนขึ้นด้วยภาษาซี

ภาษาซีมีประสิทธิภาพสำหรับคอมพิวเตอร์เพื่องานคำนวณและวิทยาศาสตร์เนื่องจากความสั้นเป็ ลืองต่ำ ธรรมชาติของภาษาระดับต่ำ ธรรมชาติของภาษาที่ถูกแปล และมีส่วนคณิตศาสตร์ที่ดีในไลบรารีมาตรฐานตัวอย่างของการใช้ภาษาซีในงานคำนวณและวิทยาศาสตร์ เช่นจีเอ็มพี ไลบรารีวิทยาศาสตร์ของกนู แมเทอแมติกา แมคแอลับ และแซส

ภาษาซีบางครั้งใช้เป็นภาษาระหว่างกลางในการทำให้เกิดผลของภาษาอื่นแนวคิดนี้อาจใช้เพื่อความสะดวกต่อการเคลื่อนย้ายโดยให้ภาษาซีเป็นภาษาระหว่างกลางซึ่งไม่จำเป็นต้องพัฒนาตัวสร้างรหัสแบบเจาะจงเครื่องตัวแปลโปรแกรมที่ใช้ภาษาซีในทางนี้เช่น บิตชี เกมบิต จีเอชซี สควิก และวาลา เป็นต้น อย่างไรก็ตามภาษาซีถูกออกแบบมาเพื่อเป็นภาษาเขียนโปรแกรม ไม่ใช่ภาษาเป้าหมายของตัวแปลโปรแกรมจึงเหมาะสมน้อยกว่าสำหรับการใช้เป็นภาษาระหว่างกลางด้วยเหตุผลนี้นำไปสู่การพัฒนาภาษาระหว่างกลางที่มีพื้นฐานบนภาษาซีเช่น ภาษาซีโมแนสโมแนส

ผู้ใช้งานปลายใช้ภาษาซีอย่างแพร่หลายเพื่อสร้างแอปพลิเคชันของผู้ใช้เองแต่เมื่อแอปพลิเคชันใหญ่ขึ้นการพัฒนาเช่นนั้นมักจะย้ายไปทำในภาษาอื่นที่พัฒนามาด้วยกันเช่นภาษาซีพลัสพลัส ภาษาซีชาร์ป ภาษาวิชวลเบสิก เป็นต้น

### 2.3.9 วากยสัมพันธ์

รหัสต้นฉบับของภาษาซีมีรูปแบบอิสระ ซึ่งสามารถใช้อักขระช่องว่างเท่าใดก็ได้ในรหัสมากกว่าที่จะถูกจำกัดด้วยคอลัมน์หรือบรรทัดข้อความอย่าง ภาษาฟอร์แทรน 77 ข้อความหมายเหตุจะปรากฏระหว่างตัวกัน /\* และ \*/ (แบบดั้งเดิม) หรือตามหลัง //จนกว่าจะจบบรรทัด (ภาษาซี99 เป็นต้นไป)

รหัสต้นฉบับแต่ละไฟล์ประกอบด้วยการประกาศและการนิยามฟังก์ชันต่าง ๆ และการนิยามฟังก์ชันก็ประกอบด้วยการประกาศและข้อความสั่งต่าง ๆ ภายในอีกด้วยการประกาศอาจกำหนดชนิดข้อมูลใหม่โดยใช้คำหลักเช่น struct , union และ enum หรือกำหนดค่าของชนิดข้อมูลและอาจสงวนเนื้อที่สำรองให้กับตัวแปรใหม่ โดยการเขียนชื่อของชนิดข้อมูลตามด้วยชื่อตัวแปร คำหลักอาทิ char และ int

เป็นชนิดข้อมูลพื้นฐานที่มาพร้อมกับภาษา ส่วนต่าง ๆ ของรหัสถูกคลุมด้วยวงเล็บปีกกา { กับ } เพื่อจำกัค ข อ บ เ ข ต ข อ ง ก า ร ป ร ะ ก า ศ และเพื่อกระทำเสมือนข้อความสิ่งเดียวสำหรับโครงสร้างการควบคุม

ภาษาซีใช้ข้อความสั่ง (statement) ในการระบุการกระทำเช่นเดียวกับภาษาเชิงคำสั่งอื่น ข้อความสั่งที่สามัญที่สุดคือ ข้อความสั่งนิพจน์ (expression statement) ซึ่งประกอบด้วยนิพจน์ที่จะถูกนำไปประเมินค่าตามด้วยอัฒภาค ; จากผลข้างเคียงของการประเมินค่าฟังก์ชันหลายฟังก์ชันอาจถูกเรียกใช้และตัวแปรหลายตัวอาจถูกกำหนดค่าใหม่ ภาษาซีได้เตรียมข้อความสั่งสำหรับควบคุมการไหลของโปรแกรมไว้หลายข้อความซึ่งดูได้จากคำสั่งต่าง ๆ ตัวอย่างเช่น การใช้ if – else เพื่อการทำงานแบบมีเงื่อนไข และการใช้ do – while , while และ for เพื่อการทำงานแบบ เพื่อปรับเปลี่ยนการทำงานอันเป็นลำดับปกติ เป็นสิ่งที่รองรับสำหรับการเขียนโปรแกรมเชิงโครงสร้าง สำหรับข้อความสั่ง for นั้นมีนิพจน์ของการกำหนดค่าเริ่มต้น การทดสอบเงื่อนไข และการกำหนดค่ารอบใหม่ทั้งสามอย่างในตัวเอง ซึ่งสามารถละเว้นนิพจน์ใดก็ได้ ข้อความสั่ง Break และ continue สามารถใช้ภายในการทำงานแบบวนรอบเพื่อหยุดการวนรอบหรือข้ามไปยังการกำหนดค่ารอบใหม่ทันทีตามลำดับ นอกจากนี้ยังมีข้อความสั่งที่ไม่เป็นเชิงโครงสร้างคือ goto ซึ่งจะทำให้การไหลของโปรแกรมข้ามไปยังป้าย (label) ที่ตั้งชื่อไว้ทันทีภายในฟังก์ชัน ข้อความสั่ง switch และ case ใช้สำหรับพิจารณาทางเลือกของการทำงานโดยพิจารณาจากนิพจน์ที่เป็นจำนวนเต็ม

นิพจน์ต่าง ๆ สามารถใช้ตัวดำเนินการที่มีมาพร้อมกับภาษาได้หลากหลาย (ดูด้านล่าง) แต่ละอย่างมีกฎระเบียบการใช้ฟังก์ชันอาร์กิวเมนต์ของฟังก์ชันและตัวถูกดำเนินการของตัวดำเนินการส่วนใหญ่ที่จะถูกประเมินค่านั้นไม่มีการระบุลำดับ การประเมินค่าจึงอาจแทรกซ้อนกันก็ได้ อย่างไรก็ตามผลกระทบที่เกิดขึ้นทั้งหมด (รวมทั้งที่เก็บข้อมูลตัวแปร) จะปรากฏก่อนจุดลำดับ (sequence point) ถัดไป จุดลำดับนั้นคือจุดสิ้นสุดของข้อความสั่งของแต่ละนิพจน์ และจุดที่เข้าและออกจากการใช้ฟังก์ชัน จุดลำดับก็เกิดขึ้นระหว่างการประเมินค่านิพจน์ที่มีตัวดำเนินการบางชนิด (เช่น &&,||,?: และ ตัวดำเนินการเงื่อนไข ) สิ่งนี้ทำให้การปรับแต่งรหัสจุดหมายให้เหมาะสมทำได้ในระดับสูง ซึ่งไม่จำเป็นต้องให้โปรแกรมเมอร์ภาษาซีใส่ใจมากนักเพื่อให้ได้ผลลัพธ์ที่เชื่อถือได้ ในขณะที่จำเป็นสำหรับภาษาโปรแกรมอื่น

ถึงแม้ว่าวากยสัมพันธ์ของภาษาซีจะถูกเลียนแบบโดยภาษาอื่นหลายภาษาเพราะว่าความเคยชินอย่างกว้างขวาง แต่ก็ถูกวิพากษ์วิจารณ์บ่อยครั้ง ตัวอย่างเช่น เคอร์นิกันและริตชีได้กล่าวในบทนำของ เดอะซีโปรแกรมมิงแลงกวิจ ไว้ว่า "ภาษาซีก็มีคำนิยามของมันเหมือนภาษาอื่นใด ตัวดำเนินการบางตัวมีสิทธิการมาก่อนที่ผิดวากยสัมพันธ์บางส่วนสามารถทำให้ดีกว่านี้"

ปัญหาเฉพาะบางอย่างที่ควรหมายเหตุไว้มีดังนี้

2.3.9.1 ไม่มีการตรวจสอบจำนวนและชนิดของอาร์กิวเมนต์ เมื่อการประกาศฟังก์ชันมีรายการพารามิเตอร์ว่าง (สิ่งนี้เพื่อความเข้ากันได้ย้อนหลังกับภาษาเคแอลดีอาร์ซี ซึ่งไม่มีโปรโตไทป์)

2.3.9.2 ทางเลือกที่น่าสงสัยของสิทธิการมาก่อนของตัวดำเนินการ ดังที่กล่าวถึงโดยเคอร์นิกันและริตชีข้างต้น เช่น  $==$  ที่วางอยู่ติดกับ  $\&$  และ  $!$  ในนิพจน์ดังตัวอย่าง  $x \& 1 = 0$  ตัวดำเนินการ  $=$  จะทำก่อนซึ่งไม่ใช่ผลที่คาดไว้ จำเป็นต้องใส่วงเล็บเพิ่ม  $(x \& 1 == 0)$  เพื่อให้  $\&$  ทำก่อนตามต้องการ

2.3.9.3 ตัวดำเนินการ  $=$  ซึ่งใช้แสดงภาวะเท่ากันในคณิตศาสตร์แต่ในภาษาซีใช้เพื่อกำหนดค่าของตัวแปร โดยใช้ตามแบบที่มีอยู่ก่อนในภาษาฟอร์แทรนภาษาพีแอล/วันและภาษาบสิก ไม่เหมือนภาษาอัลกอลและภาษาต่อของมัวร์ริตชีตั้งใจเลือกรูปแบบนี้ด้วยเหตุผลหลักว่าอาร์กิวเมนต์ของการกำหนดค่าเกิดขึ้นบ่อยกว่าการเปรียบเทียบ

2.3.9.4 ความคล้ายกันของตัวดำเนินการกำหนดค่าและการเปรียบเทียบภาวะเท่ากัน และทำให้เกิดความผิดพลาดจากการใช้เครื่องหมายผิดได้ง่าย ในหลายกรณีเครื่องหมายถูกใช้ในบริบทของอีกอันหนึ่ง โดยไม่มีความผิดพลาดขณะแปลแม้ว่าตัวแปลโปรแกรมปกติจะสร้างข้อความเตือนขึ้นมาตัวอย่างเช่น นิพจน์เงื่อนไขภายใน  $\text{if}(a=b)$  จะเป็นจริงถ้า  $a$  มีค่าไม่เป็นศูนย์หลังจากการกำหนดค่าอย่างใดก็ตาม ข้อบกพร่องนี้อาจมีประโยชน์สำหรับการเขียนรหัสอย่างย่อในบางกรณี

2.3.9.5 การขาดตัวดำเนินการเติมกลางสำหรับวัตถุซับซ้อนหลายชนิดโดยเฉพาะการดำเนินการสายอักขระทำให้โปรแกรมที่ขึ้นอยู่กับการดำเนินการเหล่านี้มีขนาดใหญ่กว่าที่ควรเป็น (เพราะต้องสร้างฟังก์ชันขึ้นเอง) และทำให้รหัสอ่านยากขึ้นด้วย

2.3.9.6 รูปแบบของการประกาศที่บางครั้งไม่เป็นไปตามสามัญสำนึกโดยเฉพาะตัวชี้ฟังก์ชันแนวคิดของริตชีคือการประกาศตัวระบุในบริบทที่สัมพันธ์กับการใช้งานของมัน

2.3.10 ตัวดำเนินการ



ภาษาซีรองรับตัวดำเนินการหลายประเภท  
ซึ่งเป็นสัญลักษณ์ที่ใช้ในนิพจน์เพื่อระบุการจัดการที่จะถูกทำให้เกิดผล  
ระหว่างการประเมินค่าของนิพจน์นั้น ภาษาซีมีตัวดำเนินการต่อไปนี้

- 2.3.10.1 พีชคณิต (+, -, \*, /, %)
- 2.3.10.2 การกำหนดค่า (=)
- 2.3.10.3 การกำหนดค่าแต่งเติม (+=, -=, \*=, /=, %=, &=, <<=, >>=)
- 2.3.10.4 ตรรกะระดับบิต (&)
- 2.3.10.5 การเลื่อนระดับบิต (<<, >>)
- 2.3.10.6 ตรรกะแบบบูล (!, &&, ||)
- 2.3.10.7 การประเมินค่าเชิงเงื่อนไข (?:)
- 2.3.10.8 การทดสอบภาวะเท่ากัน (==, !=)
- 2.3.10.9 การรวมอาร์กิวเมนต์ฟังก์ชัน (( ))
- 2.3.10.10 การเพิ่มค่าและการลดค่า (++ , --)
- 2.3.10.11 การเลือกสมาชิกในวัตถุ (., ->)
- 2.3.10.12 ขนาดของวัตถุ (sizeof)
- 2.3.10.13 ความสัมพันธ์เชิงอันดับ (<, <=, >, >=)
- 2.3.10.14 การอ้างอิงและการถูกอ้างอิง (&, \*, [ ])
- 2.3.10.15 การลำดับ (,)
- 2.3.10.16 การจัดกลุ่มนิพจน์ย่อย (( ))
- 2.3.10.17 การแปลงชนิดข้อมูล (( ))

#### 2.3.11 การแปลงจำนวนเต็ม จำนวนจุดลอยตัว และการปัดเศษ

วากยสัมพันธ์ของการแปลงชนิดข้อมูลสามารถใช้แปลงค่าต่าง

ๆ

ระหว่างชนิดข้อมูลจำนวนเต็มและจำนวนจุดลอยตัว (จำนวนทศนิยม)

หรือระหว่างจำนวนเต็มสองจำนวน หรือระหว่างจำนวนจุดลอยตัวสองจำนวนที่มีขนาดแตกต่างกัน

ตัวอย่างเช่น (long int)sqrt(1000.0), (double)(256\*256) หรือ (float)sqrt(1000.0) เป็นต้น

การแปลงชนิดข้อมูลเป็นภาวะปริยายในหลายบริบท อาทิเมื่อกำหนดค่าให้กับตัวแปรหรือพารามิเตอร์ของฟังก์ชัน หรือเมื่อใช้จำนวนจุดลอยตัวเป็นดัชนีของเวกเตอร์ หรือในการดำเนินการทางเลขคณิตที่มีตัวถูกดำเนินการเป็นข้อมูลคนละชนิดกัน

การแปลงค่าระหว่างจำนวนเต็มและจำนวนจุดลอยตัวโดยทั่วไปจะเกิดการเปลี่ยนแปลงการเข้ารหัสระดับบิตไปยังขอบเขตที่เป็นไปได้เพื่อสว่นค่าจำนวนของตัวถูกดำเนินการนั้นไม่เหมือนกับ การแปลงชนิดข้อมูลอื่น (ซึ่งการเข้ารหัสระดับบิตของตัวถูกดำเนินการจะถูกตีความใหม่ตามชนิดเป้าหมายเพียงเท่านั้น) โดยเงื่อนไขว่าจะอยู่ข้างใดข้างหนึ่ง การแปลงชนิดข้อมูลจากจำนวนเต็มไปเป็นจำนวนจุดลอยตัวจะคงไว้ซึ่งค่าจำนวนได้อย่างถูกต้องเว้นแต่ถ้าจำนวนบิตในชนิดเป้าหมายมีไม่เพียงพอ กรณีดังกล่าวจะทำให้บิตที่มีนัยสำคัญน้อยที่สุดสูญหายไป

ส่วนการแปลงชนิดข้อมูลจากจำนวนจุดลอยตัวไปเป็นจำนวนเต็มจะเกิดการตัดค่าหลังจุดทศนิยมอย่างหลีกเลี่ยงไม่ได้ (ค่าถูกปัดเศษเข้าหาศูนย์) สำหรับการปัดเศษชนิดอื่น ภาษซี99ได้ระบุไว้แล้วในฟังก์ชันดังนี้ (ใน `<math.h>`)

`round()`: ปัดเศษไปยังจำนวนเต็มที่ใกล้ที่สุด

`rint()`, `nearbyint()`: ปัดเศษตามทิศทางของจำนวนจุดลอยตัวปัจจุบัน

`ceil()`: ค่าจำนวนเต็มน้อยสุดที่ไม่น้อยกว่าอาร์กิวเมนต์ (ปัดขึ้น) ดูเพิ่มที่ฟังก์ชันเพดาน

`floor()`: ค่าจำนวนเต็มมากที่สุดที่ไม่มากกว่าอาร์กิวเมนต์ (ปัดลง) ดูเพิ่มที่ฟังก์ชันพื้น

`trunc()`: ปัดเศษเข้าหาศูนย์ (เหมือนกับการแปลงชนิดข้อมูลเป็นจำนวนเต็ม)

ฟังก์ชันทั้งหมดนี้รับอาร์กิวเมนต์ `double` และคืนค่าเป็น `double` ซึ่งต่อจากนี้ก็อาจแปลงชนิดข้อมูลเป็นจำนวนเต็มอีกทีหากจำเป็น

การแปลงชนิดข้อมูลจาก `float` ไปเป็น `double` จะคงไว้ซึ่งค่าจำนวนได้อย่างถูกต้อง ในขณะที่การแปลงกลับ ค่าจะถูกปัดเศษ ซึ่งมักเป็นการปัดเศษเข้าหาศูนย์ เพื่อให้พอดีกับจำนวนบิตที่น้อยลง (เนื่องจาก `float` ก็มีช่วงเลขชี้กำลังที่น้อยกว่าด้วยการแปลงชนิดข้อมูลอาจให้ผลเป็นค่าอนันต์แทน) ตัวแปลโปรแกรมบางโปรแกรมจะแปลงค่าของ `float` ไปเป็น `double` โดยเบื้องหลังในบางบริบทเช่น พารามิเตอร์ของฟังก์ชันที่ประกาศเป็น `float` ตามความเป็นจริงอาจส่งค่าเป็น `double` ก็ได้

เครื่องมือที่ทำตามมาตรฐานจำนวนจุดลอยตัวของIEEE เหตุการณ์การปัดเศษบางเหตุการณ์มีผลมาจากสถานะการปัดเศษปัจจุบัน (ได้แก่การปัดเศษเลขคู่ การปัดเศษขึ้น การปัดเศษลง และการปัดเศษเข้าหาศูนย์) ซึ่งอาจเรียกดูหรือตั้งค่าสถานะโดยใช้ฟังก์ชัน `fegetround()/fesetround()` ที่นิยามไว้ใน `<fenv.h>`

### 2.3.12 ตัวอย่างโปรแกรม “Hello World”

ตัวอย่างโปรแกรม “Hello World” ซึ่งปรากฏอยู่ในหนังสือ เดอะซีโปรแกรมมิงแลงกวิจ ที่พิมพ์ครั้งแรกกลายมาเป็นตัวแบบของโปรแ

กรมเกรียนนำในตำราการเขียนโปรแกรมส่วนใหญ่หากไม่คำนึงถึงภาษาที่ใช้เขียน  
โปรแกรมดังกล่าวจะแสดงผล "hello world"  
ทางอุปกรณ์ส่งออกมาตรฐาน ซึ่งมักจะเป็นเครื่องปลายทางหรือหน่วยแสดงผลจอภาพ

รหัสโปรแกรมรุ่นดั้งเดิมเป็นดังนี้

```
main()
{
    printf("hello world\n");
}
```

และหลังจากการปรับเปลี่ยนรหัสให้เข้ากับมาตรฐาน รหัสจึงเป็นดังนี้

```
#include <stdio.h>

int main(void)
{
    printf("hello world\n");
    return 0;
}
```

บรรทัดแรกของโปรแกรมเป็นคำสั่งชี้แนะตัวประมวลผลก่อน (preprocessing directive) แสดงไว้โดย #include ทำให้ตัวประมวลผลก่อน (อันเป็นเครื่องมืออย่างแรกที่พิจารณาต้นฉบับ) นำเนื้อหาข้อความทั้งหมดของไฟล์ส่วนหัวมาตรฐาน stdio.h เข้ามาแทนที่บรรทัดนั้น ซึ่งไฟล์ดังกล่าวมีการประกาศฟังก์ชันสำหรับอุปกรณ์นำเข้าและส่งออกมาตรฐาน อาทิ printf วงเล็บแหลมที่คลุมชื่อไฟล์ stdio.h (ซึ่งความจริงคือเครื่องหมายน้อยกว่า-มากกว่า) ปีกว่า และน้อยกว่า-มากกว่า a stdio.h ถูกกำหนดที่ตั้งโดยใช้กลยุทธการค้นหาที่ให้ความสำคัญต่อไฟล์ส่วนหัวมาตรฐาน มากกว่าไฟล์ส่วนหัวอื่นที่มีชื่อเดียวกัน อันุญาตกล่าวได้ว่าในกรณีที่ต้องการนำไฟล์ส่วนหัวที่อยู่ใกล้เคียงหรือเจาะจงโครงการเข้ามารวม

บรรทัดถัดมาเป็นกฏนิยามฟังก์ชันชื่อว่า main ฟังก์ชัน main เป็นฟังก์ชันที่มีจุดประสงค์พิเศษในโปรแกรมภาษาซี สภาพแวดล้อมขณะทำงานจะเรียกใช้ฟังก์ชัน main เพื่อเริ่มต้นการทำงานโปรแกรม ตัวระบุชนิด int เป็นตัวแสดงว่าค่าส่งคืนที่ถูกส่งคืนโดยตัวที่เรียกใช้ (กรณีนี้คือสภาพแวดล้อมขณะทำงาน) จะเป็นจำนวนเต็มค่าหนึ่ง อันเป็นผลจากการประเมินค่าของฟังก์ชัน main ค่าหลัก void ในรายการพารามิเตอร์แสดงว่าฟังก์ชัน main ไม่ต้องใช้อาร์กิวเมนต์

วงเล็บปีกกาเปิดหมายถึงจุดเริ่มต้นของการนิยามฟังก์ชัน main บรรทัดถัดมาเป็นการเรียกใช้ฟังก์ชันที่ชื่อว่า printf ซึ่งประกาศไว้ใน stdio.h และจัดเตรียมขึ้นจากไลบรารีของระบบ ในการเรียกใช้ครั้งนี้ ฟังก์ชัน printf จะถูกผ่านค่าด้วยอาร์กิวเมนต์หนึ่งตัวคือตำแหน่งหน่วยความจำของอักขระตัวแรกในสายอักขระ “Hello World”/n สายอักขระดังกล่าวคือแถวลำดับที่ไม่มีชื่ออันประกอบด้วยชนิดข้อมูล char จะถูกสร้างขึ้นโดยอัตโนมัติโดยตัวแปลโปรแกรม และแถวลำดับจะมีอักขระค่าศูนย์ (null) เป็นสิ่งที่บ่งบอกจุดสิ้นสุดของสายอักขระ (printf จำเป็นต้องทราบสิ่งนี้) /n ที่ปรากฏในสายอักขระคือลำดับการหนี (escape sequence) ภาษาซีจะตีความว่าเป็นอักขระขึ้นบรรทัดใหม่ (newline) ซึ่งจะทำให้อุปกรณ์ส่งออกทราบว่าถึงจุดสิ้นสุดของบรรทัดปัจจุบัน ค่าส่งคืนจากฟังก์ชัน printf คือชนิด int แต่มันถูกละทิ้งไปอย่างเจียม ๆ เนื่องจากไม่มีการใช้ (โปรแกรมที่ระมัดระวังมากกว่าอาจทดสอบค่าส่งคืนเพื่อพิจารณาว่าผลจากการทำงานของฟังก์ชัน printf สำเร็จหรือไม่) อัมภาค ; เป็นจุดสิ้นสุดข้อความสั่ง

ข้อความสั่ง return เป็นการสิ้นสุดการทำงานของฟังก์ชัน main และทำให้ฟังก์ชันส่งกลับเป็นจำนวนเต็มค่า 0 ซึ่งสภาพแวดล้อมขณะทำงานจะตีความว่าเป็นรหัสออกจากโปรแกรมที่แสดงว่าการทำงานประสบความสำเร็จ

วงเล็บปีกกาปิดหมายถึงจุดสิ้นสุดของการนิยามฟังก์ชัน main

### 2.3.12 ชนิดข้อมูล

ภาษาซีมีระบบชนิดตัวแปรแบบไม่เคร่งครัด ซึ่งมีความคล้ายคลึงบางประการร่วมกับภาษาอื่น ๆ ภาษาซีมีชนิดตัวแปรที่เตรียมไว้แล้วสำหรับจำนวนเต็มหลายขนาดแบบทั้งมีเครื่องหมายและไม่มีเครื่องหมาย จำนวนจุดลอยตัว ตัวอักขระ และชนิดข้อมูลแจกแจง (enum) ในภาษาซี 99 ได้เพิ่มชนิดตัวแปรแบบบูลเข้าไปด้วย ภาษาซีก็ยังมีชนิดตัวแปรที่รับทอดมาด้วยเช่นแถวลำดับ ตัวชี้ระเบียน (struct) และยูเนียน (union)

ภาษาซีมักใช้กับการเขียนโปรแกรมระบบในระดับต่ำ ซึ่งอาจหลบเลี่ยงการใช้ระบบชนิดตัวแปรเมื่อจำเป็น ตัวแปลโปรแกรมจะพยายามทำให้แน่ใจว่าชนิดตัวแปรถูกใช้อย่างถูกต้องในนิพจน์ส่วนใหญ่ แต่โปรแกรมเมอร์ก็สามารถลบการตรวจสอบเช่นนั้นได้หลายทาง อาทิ การโยนชนิดข้อมูล (type cast) เพื่อแปลงค่าจากชนิดหนึ่งไปเป็นชนิดหนึ่งอย่างชัดเจน หรือการใช้ตัวชี้หรือยูเนียนเพื่อแปลความหมายบิตของค่าที่อยู่ภายในไปเป็นอีกชนิดหนึ่ง

### 2.3.13 ตัวชี้

ภาษาซีรองรับการใช้อ้างอิงอย่างง่ายชนิดหนึ่ง ที่เก็บบันทึกที่อยู่หรือตำแหน่งของวัตถุหรือฟังก์ชันในหน่วยความจำ ตัวชี้สามารถอ้างอิงกลับ (dereference) เพื่อเข้าถึงข้อมูลที่บันทึกในตำแหน่งที่ถูกชี้อยู่ หรือเพื่อเรียกใช้ฟังก์ชันที่ถูกชี้อยู่ ตัวชี้สามารถจัดดำเนินการกำหนดค่าและเลขคณิตของตัวชี้ได้ด้วย ค่าของตัวชี้ขณะโปรแกรมทำงาน มักจะเป็นตำแหน่งมาตรฐานในหน่วยความจำ (ซึ่งอาจเสริมด้วยค่าออฟเซตในหน่วยเวิร์ด) แต่เนื่องจากตัวชี้มีการระบุชนิดตามข้อมูลที่ชี้ไป ตัวแปรโปรแกรมจึงสามารถตรวจสอบชนิดตัวแปรในนิพจน์ต่าง ๆ รวมทั้งตัวชี้ด้วยกันเองขณะแปลได้ เลขคณิตของตัวชี้จะแปรสัดส่วนของขนาดโดยอัตโนมัติตามชนิดข้อมูลที่ชี้ไป (ดูเพิ่มที่ส่วนความใช้แทนกันไ้ระหว่างตัวชี้และแถวลำดับ) จุดประสงค์ของการใช้ตัวชี้มีหลากหลายในภาษาซี เช่น สายอักขระมักจัดดำเนินการโดยใช้ตัวชี้ไปยังแถวลำดับของตัวอักขระ การจัดสรรหน่วยความจำพลวัต (dynamic memory allocation) สามารถกระทำได้ด้วยตัวชี้ ชนิดข้อมูลชนิดอื่นเช่น ต้นไม้ ปกติจะถูกพัฒนาขึ้นโดยการจัดสรรวัตถุ struct โดยพลวัต ซึ่งเชื่อมโยงแต่ละหน่วยเข้ากันด้วยตัวชี้ ตัวชี้ของฟังก์ชันใช้เพื่อการเรียกกลับ (callback) สำหรับชุดคำสั่งจัดการเหตุการณ์ เป็นต้น

ตัวชี้ว่าง (null pointer) คือตัวชี้ที่ชี้ไปยังตำแหน่งที่ใช้งานไม่ได้ ซึ่งจะมีค่าเป็น 0 การอ้างอิงกลับของตัวชี้ว่างจึงไม่มีความหมาย และโดยทั่วไปให้ผลเป็นข้อผิดพลาดขณะทำงาน อย่างไรก็ตามตัวชี้ว่างก็มีประโยชน์สำหรับกรณีพิเศษ เช่น ใช้เป็นจุดสิ้นสุดหน่วยสุดท้ายของรายการโยง ซึ่งหมายความว่าไม่มีตัวชี้ไปหน่วยอื่นแล้ว หรือใช้แจ้งข้อผิดพลาดจากฟังก์ชันที่คืนค่าเป็นตัวชี้ ตัวชี้ว่างในการลงรหัสมักจะนำเสนอด้วย 0 หรือ NULL

ตัวชี้วอยด์ (void\*) คือตัวชี้ของวัตถุที่ไม่ทราบชนิดตัวแปร ดังนั้นจึงสามารถใช้เป็นตัวชี้ "ทั่วไป" ก็ได้ แต่เนื่องจากขนาดและชนิดของวัตถุที่ถูกชี้ไม่เป็นที่ทราบ ตัวชี้วอยด์จึงไม่สามารถอ้างอิงกลับได้ และเลขคณิตของตัวชี้ก็ใช้กับตัวชี้วอยด์ไม่ได้ แม้ว่าตัวชี้ของวัตถุชนิดหนึ่งอาจแปลงเป็นตัวชี้ชนิดอื่นได้โดยง่าย (และในหลายบริบทก็แปลงได้อย่างคลุมเครือ)

การใช้งานตัวชี้อย่างไม่ระมัดระวังอาจเกิดอันตรายได้ เนื่องจากตัวแปรตัวชี้สามารถชี้ไปที่ตำแหน่งใดก็ได้โดยไม่มีกฎเกณฑ์ และปกติก็ไม่มีการตรวจสอบ ซึ่งอาจทำให้เกิดผลกระทบที่ไม่พึงปรารถนา ถึงแม้ตัวชี้ที่ใช้งานอย่างถูกต้องได้ชี้ไปยังตำแหน่งที่ปลอดภัยอยู่แล้ว แต่มันก็อาจถูกทำให้ชี้ไปยังตำแหน่งที่ไม่ปลอดภัยโดยการดำเนินการเลขคณิตที่ไม่ถูกต้อง หรือตัวชี้ไปยังวัตถุที่อาจเรียกคืนการจัดสรรไปแล้วแต่ถูกเรียกใช้ใหม่ (ตัวชี้อย่างหลวม dangling pointer) หรือตัวชี้ที่อาจใช้งานโดยไม่กำหนดค่าเริ่มต้น (ตัวชี้ตัวแทน wild pointer) หรือตัวชี้ที่อาจถูกกำหนดด้วยค่าที่ไม่ปลอดภัยโดยตรง ด้วยวิธีของชนิดตัวแปร ยูเนียน หรือ ผ่าน ค่า มา จาก ตัว ชี้ อื่น ที่ เสีย เป็น ต้น โดยทั่วไปภาษาซีอนุญาตให้จัดดำเนินการและแปลงชนิดตัวแปรของตัวชี้ได้ แม้ว่าตัวแปลโปรแกรมก็มีตัวเลือกสำหรับการตรวจสอบอยู่หลายระดับก็ตาม ภาษาโปรแกรมอื่นบางภาษาจัดการปัญหานี้โดยกำหนดให้ใช้ชนิดตัวแปรอ้างอิงที่เคร่งครัดมากกว่า

ภาษาซีรองรับการใช้งานตัวชี้ (pointer) ซึ่งเป็นชนิดข้อมูลสำหรับการอ้างอิงอย่างง่ายชนิดหนึ่ง ที่เก็บบันทึกที่อยู่หรือตำแหน่งของวัตถุหรือฟังก์ชันในหน่วยความจำ ตัวชี้สามารถอ้างอิงกลับ (dereference) เพื่อเข้าถึงข้อมูลที่บันทึกในตำแหน่งที่ถูกชี้อยู่ หรือเพื่อเรียกใช้ฟังก์ชันที่ถูกชี้อยู่ ตัวชี้สามารถจัดดำเนินการกำหนดค่าและเลขคณิตของตัวชี้ได้ด้วย ค่าของตัวชี้ขณะโปรแกรมทำงาน มักจะเป็นตำแหน่งมูลฐานในหน่วยความจำ (ซึ่งอาจเสริมด้วยค่าออฟเซตในหน่วยเวิร์ด) แต่เนื่องจากตัวชี้มีการระบุชนิดตามข้อมูลที่ชี้ไป ตัวแปลโปรแกรมจึงสามารถตรวจสอบชนิดตัวแปรในนิพจน์ต่างๆ รวมทั้งตัวชี้ด้วยกันเองขณะแปลได้ เลขคณิตของตัวชี้จะแปรสัดส่วนของขนาดโดยอัตโนมัติตามชนิดข้อมูลที่ชี้ไป (ดูเพิ่มที่ส่วนความใช้แทนกันได้ระหว่างตัวชี้และแถวลำดับ) จุดประสงค์ของการใช้ตัวชี้มีหลากหลายในภาษาซี เช่น สายอักขระมักจัดดำเนินการโดยใช้ตัวชี้ไปยังแถวลำดับของตัวอักขระ การจัดสรรหน่วยความจำพลวัต (dynamic memory allocation) สามารถกระทำได้ด้วยตัวชี้ ชนิดข้อมูลชนิดอื่นเช่น ต้นไม้ปกติจะถูกพัฒนาขึ้นโดยการจัดสรรวัตถุ struct โดยพลวัต ซึ่งเชื่อมโยงแต่ละหน่วยเข้ากันด้วยตัวชี้ ตัวชี้ของฟังก์ชันใช้เพื่อการเรียกกลับ (callback) สำหรับชุดคำสั่งจัดการเหตุการณ์ เป็นต้น

ตัวชี้ว่าง (null pointer) คือตัวชี้ที่ชี้ไปยังตำแหน่งที่ใช้งานไม่ได้ ซึ่งจะมีค่าเป็น 0 การอ้างอิงกลับของตัวชี้ว่างจึงไม่มีความหมาย และโดยทั่วไปให้ผลเป็นข้อผิดพลาดขณะทำงาน อย่างไรก็ตาม ตัวชี้ว่างก็มีประโยชน์สำหรับกรณีพิเศษเช่น ใช้เป็นจุดสิ้นสุดหน่วยสุดท้ายของรายการโยง ซึ่งหมายความว่าไม่มีตัวชี้ไปหน่วยอื่นแล้ว หรือใช้แจ้งข้อผิดพลาดจากฟังก์ชันที่คืนค่าเป็นตัวชี้ ตัวชี้ว่างในการลงรหัสมักจะนำเสนอด้วย 0 หรือ NULL

ตัวชี้ลอย (void \*) คือตัวชี้ของวัตถุที่ไม่ทราบชนิดตัวแปร ดังนั้นจึงสามารถใช้เป็นตัวชี้ "ทั่วไป" ก็ได้ แต่เนื่องจากขนาดและชนิดของวัตถุที่ถูกชี้ไม่เป็นที่ทราบ ตัวชี้ลอยจึงไม่สามารถอ้างอิงกลับได้ และเลขคณิตของตัวชี้ก็ใช้กับตัวชี้ลอยไม่ได้ แม้ว่าตัวชี้ของวัตถุชนิดหนึ่งอาจแปลงเป็นตัวชี้ชนิดอื่นได้โดยง่าย (และในหลายบริบทก็แปลงได้อย่างคลุมเครือ)

การใช้งานตัวชี้อย่างไม่ระมัดระวังอาจเกิดอันตรายได้ เนื่องจากตัวแปรตัวชี้สามารถชี้ไปที่ตำแหน่งใดก็ได้โดยไม่มีกฎเกณฑ์ และปกติก็ไม่มี การตรวจสอบ ซึ่งอาจทำให้เกิดผลกระทบที่ไม่พึงปรารถนา ถึงแม้ตัวชี้ที่ใช้งานอย่างถูกต้องได้ชี้ไปยังตำแหน่งที่ปลอดภัยอยู่แล้ว แต่มันก็อาจถูกทำให้ชี้ไปยังตำแหน่งที่ไม่ปลอดภัยโดยการดำเนินการเลขคณิตที่ไม่ถูกต้อง หรือตัวชี้ไปยังวัตถุที่อาจเรียกคืนการจัดสรรไปแล้วแต่ถูกเรียกใช้ใหม่ (ตัวชี้อย่างหลวม dangling pointer) หรือตัวชี้ที่อาจใช้งานโดยไม่กำหนดค่าเริ่มต้น (ตัวชี้ตัวแทน wild pointer) หรือตัวชี้ที่อาจถูกกำหนดด้วยค่าที่ไม่ปลอดภัยโดยตรง ด้วยวิธีโยนชนิดตัวแปร ยูเนียน หรือ ผ่านค่ามาจากตัวชี้อื่นที่เสียเป็นต้น โดยทั่วไปภาษาซีอนุญาตให้จัดดำเนินการและแปลงชนิดตัวแปรของตัวชี้ได้ แม้ว่าตัวแปลโปรแกรมก็มีตัวเลือกสำหรับการตรวจสอบอยู่หลายระดับก็ตาม ภาษาโปรแกรมอื่นบางภาษาจัดการปัญหานี้โดยกำหนดให้ใช้ชนิดตัวแปรอ้างอิงที่เคร่งครัดมากกว่า

#### 2.3.14 แถวลำดับ

ชนิดข้อมูลแถวลำดับ (array) ในภาษาซีแบบดั้งเดิมมีขนาดคงที่และสถิต ซึ่งจะถูกกำหนดตอนแปลโปรแกรม (ในเวลาถัดมา มาตรฐานภาษาซี 99 อนุญาตให้สร้างแถวลำดับที่มีความยาวแปรได้) อย่างไรก็ตามแถวลำดับสามารถกำหนดให้จัดสรรเนื้อที่หน่วยความจำขนาดใดก็ได้ขณะทำงาน โดยใช้ฟังก์ชัน malloc จากไลบรารีมาตรฐาน แล้วทำให้เป็นแถวลำดับ การทำให้เป็นหนึ่งเดียวยระหว่างแถวลำดับและตัวชี้ของภาษาซี ทำให้หมายความว่าแถวลำดับที่แท้จริงและแถวลำดับที่จัดสรรอย่างพลวัตเสมือนใช้แทนกันได้

เนื่องด้วยแถวลำดับเข้าถึงผ่านตัวชี้เสมอ (ในทางปฏิบัติ) การเข้าถึงแถวลำดับจึงไม่มีการตรวจสอบขนาดภายใต้แถวลำดับ แม้ว่าตัวแปลโปรแกรมอาจมีตัวเลือกสำหรับตรวจสอบขอบเขตก็ตาม การใช้งานเกินขอบเขตของแถวลำดับจึงยังคงสามารถเป็นไปได้ ซึ่งเกิดขึ้นก่อนข้างเป็นปกติในรหัสที่เขียนอย่างไม่ระมัดระวัง และนำไปสู่ผลสะท้อนกลับหลายอย่าง อาทิ การเข้าถึงหน่วยความจำที่ไม่อนุญาต การทำให้ข้อมูลผิดแปลกไป บัฟเฟอร์ส่วนล้น และสิ่งผิดปรกติขณะทำงาน

ถึงแม้ภาษาซีรองรับแถวลำดับแบบสถิต แต่ก็ไม่จำเป็นว่าดัชนีของแถวลำดับจะต้องมีผล (การตรวจสอบขอบเขต) ตัวอย่างเช่น เราสามารถลงบันทึกค่าสมาชิกตัวที่หกกลงในแถวลำดับที่มีสมาชิกห้าตัวได้ ซึ่งจะทำให้เกิดผลที่ไม่คาดคิด ความผิดพลาดเช่นนี้เรียกว่า บัฟเฟอร์ส่วนล้น (buffer overflow/ overrun) เป็นสาเหตุที่สำคัญอย่างหนึ่งของปัญหาด้านความปลอดภัย เนื่องจากเทคโนโลยีการกำจัดการตรวจสอบขอบเขต (bounds-checking elimination) ไม่มีอยู่เลยเมื่อภาษาซีถูกนิยามขึ้น การตรวจสอบขอบเขตจึงลดทอนประสิทธิภาพอย่างรุนแรง โดยเฉพาะกับการคำนวณเชิงจำนวน เมื่อสองสามปีก่อนหน้านั้น ตัวแปลภาษาฟอร์แทรนมีตัวเลือกให้เปิดหรือปิดการตรวจสอบขอบเขตได้ แต่ตัวเลือกเช่นนี้ไม่มีประโยชน์ต่อภาษาซี เพราะอาร์กิวเมนต์ของแถวลำดับถูกผ่านค่าด้วยตัวชี้ธรรมดา

ภาษาซีไม่มีข้อกำหนดพิเศษสำหรับการประกาศแถวลำดับหลายมิติ แต่อาจจะขึ้นอยู่กับการเรียกซ้ำภายในระบบชนิดตัวแปร เพื่อประกาศแถวลำดับของแถวลำดับ ซึ่งสามารถบรรลุผลสำเร็จได้เหมือนกัน ค่าดัชนีของ "แถวลำดับหลายมิติ" ที่สร้างขึ้นสามารถพิจารณาว่าเพิ่มขึ้นตามอันดับเรียงตามแถว (row-major order)

โดยปกติแถวลำดับหลายมิติถูกใช้งานในขั้นตอนวิธีเชิงจำนวนเพื่อเก็บข้อมูลเมทริกซ์ (ซึ่งประยุกต์มาจากพีชคณิตเชิงเส้นเป็นหลัก) โครงสร้างของแถวลำดับในภาษาซีเหมาะสมเป็นอย่างดีสำหรับงานนี้ แต่เนื่องจากแถวลำดับถูกผ่านค่าด้วยตัวชี้ ขอบเขตของแถวลำดับจึงต้องเป็นค่าที่ทราบและตายตัว หรือไม่เช่นนั้นก็ต้องผ่านค่าไปพร้อมกับตัวชี้ที่จำเป็นต้องทราบ นอกจากนี้ แถวลำดับของแถวลำดับที่จัดสรรขนาดแบบพลวัต ไม่สามารถเข้าถึงได้โดยใช้ดัชนีสองชั้น (ตัวอย่างกรณีนี้เช่นการจัดสรรแถวลำดับด้วย "เวกเตอร์แถว" ของตัวชี้ไปยังสตริง)

ภาษาซี 99 ได้แนะนำ "แถวลำดับความยาวแปรได้" เพิ่มเข้ามา แต่ก็ยังมีปัญหาบางประการที่เหมือนกับปัญหาแถวลำดับของภาษาซี



### 2.3.15 การใช้แทนกันได้ระหว่างตัวชี้และแวลค่าดับ

คุณลักษณะเด่นชัดของภาษาซี (ซึ่งอาจทำให้สับสนด้วย) คือการปฏิบัติต่อแวลค่าดับและตัวชี้ สัญกรณ์แวลค่าดับ  $x[i]$  สามารถใช้กับตัวชี้  $x$  ได้ โดยแปลความหมายว่าเป็นการเข้าถึงวัตถุตัวที่  $i+1$  ของวัตถุข้อมูลที่อยู่ติดกันถัดจากตำแหน่งที่ ชื่ออยู่ ซึ่งถือว่าเป็นสมาชิกตัวแรกของแวลค่าดับ ( $x[0]$ )

$x[0]$  มีความหมายเทียบเท่ากับ  $*(x+i)$  ตามรูปแบบ และเนื่องจากชนิดตัวแปรของตัวชี้เป็นที่ทราบขณะแปล ตำแหน่ง  $x + i$  ที่ชี้ไปไม่ได้หมายความว่าจากตำแหน่ง  $x$  แล้วเพิ่มไปอีก  $i$  ไบต์ แต่หมายถึงเพิ่มไปอีก ( $i$  คูณด้วยขนาดของสมาชิกที่ตำแหน่ง  $x$ ) ขนาดของสมาชิกนี้ได้มาจากการใช้ตัวดำเนินการ `sizeof` บนสมาชิกที่อ้างอิงกลับตัวใดตัวหนึ่งของ  $x$  ดังเช่น  $n = \text{sizeof } *x$  หรือ  $n = \text{sizeof } x[0]$

นอกจากนี้ในบริบทส่วนใหญ่ของนิพจน์ ชื่อของแวลค่าดับจะถูกแปลงเป็นตัวชี้ที่ชี้ไปยังสมาชิกตัวแรกของแวลค่าดับนั้น สิ่งนี้อาจเป็นได้ว่าแวลค่าดับจะไม่ถูกคัดลอกข้อมูลไปทั้งหมดเมื่อนำไปตั้งชื่ออาร์กิวเมนต์ของฟังก์ชัน แต่จะมีเพียงแค่ตำแหน่งของสมาชิกตัวแรกเท่านั้นที่ส่งผ่านไป ดังนั้นถึงแม้ว่าการเรียกใช้ฟังก์ชันในภาษาซีจะตีความว่าส่งโดยให้ค่า (pass-by-value) แต่แวลค่าดับนั้นส่งโดยอ้างอิง (pass-by-reference) ในทางปฏิบัติ

จำนวนสมาชิกของแวลค่าดับ  $X$  ที่ได้ประกาศไว้แล้ว สามารถคำนวณได้จาก  $\text{sizeof } x / \text{sizeof } x[0]$

การสาธิตอย่างหนึ่งที่น่าสนใจต่อการใช้แทนกันได้ระหว่างตัวชี้และแวลค่าดับแสดงไว้ด้านล่าง การกำหนดค่าทั้งสี่มีความหมายเทียบเท่ากันและเป็นรหัสที่ใช้งานได้ภาษาซี

```
/* x เป็นแวลค่าดับหรือตัวชี้, i เป็นจำนวนเต็ม */
```

```
x[i] = 1;      /* เทียบเท่ากับ *(x + i) */
```

```
*(x + i) = 1;
```

```
*(i + x) = 1;
```

```
i[x] = 1;      /* เทียบเท่ากับ *(i + x) */
```

แม้ว่าการกำหนดค่าทั้งสี่เทียบเท่ากัน แต่มีเพียงแบบแรกเท่านั้นที่แสดงรูปแบบการลงรหัสที่ดี กรณีอื่นอาจพบได้ในรหัสภาษาซีที่ยุ่งเหยิง

ถึงอย่างไรก็ตามแถวลำดับและตัวชี้ก็ยังมีจุดที่แตกต่างแม้ว่ามันจะเทียบเท่ากัน ตัวชี้ไปยังสมาชิกตัวแรกซึ่งแปลงมาจากแถวลำดับ ไม่มีเนื้อที่เก็บข้อมูลตำแหน่งของมันเอง ต่างจากตัวแปรตัวชี้ซึ่งมี เมื่อเป็นเช่นนั้นแล้วสิ่งที่แถวลำดับ "ชี้ไป" จึงไม่สามารถเปลี่ยนแปลงได้ และไม่สามารถกำหนดค่าใหม่ให้กับตัวแปรแถวลำดับ (ค่าต่าง ๆ ของแถวลำดับอาจคัดลอกได้ โดยใช้ฟังก์ชัน memcpy เป็นต้น)

### 2.3.16 การจัดการหน่วยความจำ

ฟังก์ชันการทำงานหนึ่งที่สำคัญที่สุดของภาษาโปรแกรมคือการให้บริการการจัดการหน่วยความจำและวัตถุที่บันทึกอยู่ในหน่วยความจำ ภาษาซีมีสามแนวทางที่ต่างกันเพื่อจัดสรรหน่วยความจำสำหรับวัตถุ

|  |    |     |   |
|--|----|-----|---|
| 2.   | 3. | 16. | 1 |
| การจัดสรรหน่วยความจำสถิต ที่ว่างสำหรับวัตถุในรหัสฐานสองซึ่งเว้นไว้ขณะแปลโปรแกรม        |    |     |   |
| วัตถุเหล่านี้มีอายุขัย (extent) ครบเท่าที่รหัสฐานสองที่มีวัตถุนั้นบรรจุลงในหน่วยความจำ |    |     |   |

|  |    |     |   |
|--|----|-----|---|
| 2.   | 3. | 16. | 2 |
| การจัดสรรหน่วยความจำอัตโนมัติ วัตถุชั่วคราวสามารถเก็บบันทึกในกองซ้อน (stack) และที่ว่างนี้จะถูกเรียกคืนและใช้ใหม่หลังจากวัตถุที่ประกาศเลิกการทำงานโดยอัตโนมัติ |    |     |   |

2. 3. 16. 3 การจัดสรรหน่วยความจำพลวัต บล็อกต่าง ๆ ของหน่วยความจำในขนาดที่ต้องการสามารถร้องขอได้ขณะทำงาน โดยใช้ฟังก์ชันไลบรารีอาทิ malloc จ อ ง เ นื อ ที่ ห นั ว ย ค ว า ม จ ำ ที่ เ รีย ก ว่า ฮี ป ( heap) บล็อกเหล่านี้คงอยู่จนกว่าจะถูกเรียกคืนเพื่อใช้ใหม่โดยใช้ฟังก์ชัน free ในภายหลังแนวทางสามอย่างนี้เหมาะสมในสถานการณ์และข้อแลกเปลี่ยนที่ต่างกันไปด้วย เช่น การจัดสรรหน่วยความจำสถิตไม่มีการดำเนินงานสิ้นเปลือง (overhead) เพื่อการจัดสรร การจัดสรรหน่วยความจำอัตโนมัติอาจมีการสิ้นเปลืองน้อย และการจัดสรรหน่วยความจำพลวัตอาจเป็นไปได้ว่ามีความสิ้นเปลืองอย่างมากทั้งการจัดสรรและการเรียกคืน ในทางตรงข้ามที่ว่างในกองซ้อนโดยทั่วไปมีขนาดจำกัดและไม่คงทนถาวรไปกว่าหน่วยความจำแบบสถิตหรือที่ว่างในฮีป และการจัดสรรหน่วยความจำพลวัตสามารถจัดสรรวัตถุที่ทราบขนาดเฉพาะขณะทำงานได้ โปรแกรมภาษาซีส่วนใหญ่จึงใช้งานทั้งสามแนวทางอย่างกว้างขวาง

การจัดสรรมักให้ความสำคัญแก่แบบอัตโนมัติหรือแบบสถิตมากกว่า เพราะตัวแปลโปรแกรมเป็นส่วนจัดการหน่วยเก็บบันทึก ทำให้โปรแกรมเมอร์ไม่ต้องจัดสรรและเรียกคืนหน่วยเก็บบันทึกจุกจิกด้วยตนเองซึ่งเสี่ยงต่อการเกิดข้อผิดพลาด อย่างไรก็ตามโครงสร้างข้อมูลหลายชนิดสามารถขยายขนาดได้ในขณะทำงาน

และเนื่องจากการจัดสรรสถิติ (และการจัดสรรอัตโนมัติในภาษาซี89 และซี90) จะต้องมีการตายตัวขณะแปลโปรแกรม หลายสถานการณ์จึงจำเป็นต้องใช้การจัดสรรพลวัตก่อนที่จะมีมาตรฐานซี99 แถวลำดับความยาวแปรได้เป็นตัวอย่างปัญหาหนึ่งของกรณีนี้

วัตถุที่จัดสรรแบบอัตโนมัติและพลวัตจะถูกกำหนดค่าเริ่มต้นถ้าได้ระบุไว้ หรือ มิฉะนั้น มัน จะ มี ค่า ที่ ไม่ แน่ น อน (ไม่ว่ารูปแบบรหัสฐานสองบนหน่วยเก็บบันทึกจะเป็นอะไรก็ตาม ซึ่งอาจไม่เป็นค่าที่ใช้งานได้สำหรับชนิดตัวแปรนั้น) ถ้าโปรแกรมพยายามเข้าถึงค่าที่ไม่กำหนดนี้ ผลลัพธ์จะไม่สามารถนิยามได้ ตัวแปลโปรแกรมสมัยใหม่หลายโปรแกรมพยายามตรวจสอบและแจ้งเตือนเกี่ยวกับปัญหานี้ แต่ก็เกิดทั้งผลบวกและผลลบ

ปัญหาอีกประการหนึ่งก็คือ การจัดสรรหน่วยความจำในซีจำเป็นต้องกระทำพร้อมกับการใช้งานจริงด้วยตนเองในโปรแกรมใด ๆ ก็ตาม เพื่อให้มันสามารถนำกลับมาใช้ใหม่มากที่สุดเท่าที่เป็นไปได้ ตัวอย่างเช่น ถ้ามีตัวชี้ไปยังฮีปที่ถูกจัดสรรนอกขอบเขต หรือค่าของตัวชี้ถูกเขียนทับก่อนเรียกใช้ free จะทำให้หน่วยความจำที่ตำแหน่งนั้นไม่สามารถเรียกคืนเพื่อใช้ใหม่ภายหลังและสูญเสียไปกับโปรแกรม อันเป็นปรากฏการณ์ที่เรียกว่า หน่วยความจำรั่ว (memory leak) ในทางกลับกัน การปลดปล่อยหน่วยความจำเร็วเกินไปแล้วยังคงใช้งานอยู่ซึ่งเป็นไปได้ แต่เนื่องจากระบบจัดสรรหน่วยความจำสามารถจัดสรรอีกครั้งหรือใช้หน่วยความจำที่ถูกทำให้ว่าง พฤติกรรมที่คาดเดาไม่ได้ก็อาจเกิดขึ้น โดยปกติอาการจะปรากฏในส่วน of โปรแกรมที่อยู่ไกลจากจุดที่ทำให้เกิดความผิดพลาดจริง ทำให้ตรวจสอบแก้ปัญหาได้ อย่างยากลำบาก ปัญหาเช่นนี้ได้รับการปรับปรุงแก้ไขในภาษาโปรแกรมที่มีการเก็บกวาดข้อมูลขยะอัตโนมัติ

### 2.3.17 ไลบรารี

ภาษาซีใช้ไลบรารีเป็นวิธีการหลักสำหรับส่วนขยาย ไลบรารีคือกลุ่มของฟังก์ชันที่บรรจุอยู่ในไฟล์เดียวกันโดย "ถาวร" ไลบรารีแต่ละชนิดจะมีไฟล์ส่วนหัว ซึ่งรวบรวมต้นแบบ (prototype) ตามฟังก์ชันที่มีอยู่ในไลบรารีซึ่งอาจถูกเรียกใช้โดยโปรแกรม และมีการประกาศชนิดข้อมูลพิเศษและสัญลักษณ์แมโครที่ใช้ในฟังก์ชันเหล่านั้น

โปรแกรมจะต้องรวมไฟล์ส่วนหัวนี้เข้าไปเพื่อใช้งานไลบรารี และไลบรารีจะต้องเชื่อมโยงกับโปรแกรม ซึ่งในหลายกรณีอาจต้องใช้ตัวบ่งชี้คอมไพเลอร์ (compiler flag) (เช่น -lm สำหรับไลบรารีคณิตศาสตร์เป็นต้น)

ไลบรารีสามัญที่สุดคือไลบรารีมาตรฐานของภาษาซี ซึ่งระบุไว้โดยมาตรฐานไอโซและแอนซี ซี และติดมากับทุกโปรแกรมที่พัฒนาด้วยภาษาซี (ส่วนการพัฒนาบนสภาพแวดล้อมแบบฝังตัวอาจมีไลบรารีมาตรฐานเพียงส่วนย่อยส่วนหนึ่ง) ไลบรารีนี้รองรับกระแสข้อมูลรับเข้าและส่งออกการจัดสรรหน่วยความจำ คณิตศาสตร์ สายอักขระ และค่าของเวลา

ไลบรารีสามัญอีกกลุ่มหนึ่งเป็นฟังก์ชันที่เจาะจงใช้กับโปรแกรมที่ทำงานบนระบบปฏิบัติการยูนิกซ์หรือคล้ายยูนิกซ์ โดยเฉพาะฟังก์ชันที่มีส่วนต่อประสานเข้ากับเคอร์เนล ฟังก์ชันเหล่านี้ได้ให้รายละเอียดไว้ในมาตรฐานหลากหลายเช่นโพสซิกซ์หรือข้อกำหนดคุณลักษณะยูนิกซ์เชิงเดี่ยว (Single UNIX Specification)

เนื่องด้วยโปรแกรมหลายโปรแกรมถูกเขียนขึ้นด้วยภาษาซี ไลบรารีอื่น ๆ ที่หลากหลายในวงกว้างก็มีเช่นกัน บ่อยครั้งที่ไลบรารีเหล่านั้นเขียนด้วยภาษาซี เพราะตัวแปลภาษาซีจะจัดสร้างรหัสวัตถุ (object code) ที่มีประสิทธิภาพ จากนั้นโปรแกรมเมอร์จะสร้างส่วนต่อประสานไปยังไลบรารี จึงทำให้ภาษาระดับที่สูงกว่าอย่างภาษาจาวา ภาษาเพิร์ล และภาษาไพทอน สามารถใช้งานรูทีนในรหัสวัตถุได้

### 2.3.18 เครื่องมือที่ใช้กับภาษา

เครื่องมือหลายอย่างถูกสร้างขึ้นเพื่อช่วยเหลือโปรแกรมเมอร์ภาษาซี เพื่อหลีกเลี่ยงปัญหาบางประเภทที่มากับภาษา เช่นข้อความสั่งที่มีพฤติกรรมไม่นิยาม หรือข้อความสั่งที่ปฏิบัติไม่ดีซึ่งอาจส่งผลให้เกิดพฤติกรรมที่ไม่ตั้งใจหรือความผิดพลาดขณะทำงาน

การตรวจสอบแก้ไขรหัสต้นฉบับอัตโนมัติเป็นประโยชน์สำหรับทุกภาษา และภาษาซีก็มีเครื่องมือเหล่านั้นเช่นกัน เช่น lint การใช้ lint โดยปกติเพื่อตรวจจับรหัสที่น่าสงสัยเมื่อโปรแกรมเขียนขึ้นเป็นครั้งแรก เมื่อโปรแกรมผ่านการตรวจสอบจาก lint แล้ว มันจึงจะถูกแปลด้วยตัวแปลภาษาซี ตัวแปลภาษาหลายตัวก็สามารถเลือกได้เพื่อแจ้งเตือนเกี่ยวกับโครงสร้างที่ถูกต้องตามวากยสัมพันธ์แต่อาจเกิดความผิดพลาดได้จริง มีสราซี เป็นกลุ่มแนวทางที่มีกรรมสิทธิ์เพื่อการหลีกเลี่ยงรหัสที่น่าสงสัยเช่นนั้น ซึ่งพัฒนาขึ้นสำหรับระบบฝังตัว

นอกจากนี้ยังมีตัวแปลโปรแกรม ไบรารี และกลไกระดับระบบปฏิบัติการ เพื่อการตรวจสอบขอบเขตของแถวลำดับ การตรวจจับบัฟเฟอร์ส่วนเกิน การทำให้เป็นอนุกรม (serialization) และการเก็บกวาดข้อมูลขยะอัตโนมัติ ซึ่งมีใช้ส่วนหนึ่งที่เป็นมาตรฐานของภาษาซี

เครื่องมืออื่นอย่างเช่น เพียวริฟาย แวลกรินด์ และการเชื่อมโยงกับไลบรารีที่มีฟังก์ชันจัดสรรหน่วยความจำแบบพิเศษ สามารถช่วยเปิดเผยข้อผิดพลาดในหน่วยความจำขณะทำงานได้

### 2.3.19 ภาษาที่เกี่ยวข้อง

ภาษาซีมีอิทธิพลต่อภาษาอื่นในยุคหลังทั้งในทางตรงและทางอ้อมเช่น ภาษาจาวา ภาษาเพิร์ล ภาษาพีเอชพี จาวาสคริปต์ ภาษาแอลพีซี ภาษาซีชาร์ป และซีเชลล์ของยูนิกซ์ อิทธิพลที่แพร่หลายมากที่สุดคือรูปแบบวากยสัมพันธ์ ทุกภาษาที่กล่าวมาได้รวมวากยสัมพันธ์ของข้อความตั้งกับนิพจน์ของภาษาซี พร้อมทั้งระบบชนิดตัวแปร อันเป็นตัวแทนข้อมูลและ/หรือโครงสร้างโปรแกรมขนาดใหญ่ที่ต่างไปจากของภาษาซี ซึ่งบางครั้งก็ต่างกันอย่างมาก

เมื่อแนวคิดภาษาเชิงวัตถุเป็นที่นิยม ภาษาซีพลัสพลัสและภาษาอ็อบเจกทีฟ-ซีเป็นส่วนขยายที่แตกต่างกันของภาษาซีที่ให้ความสามารถเชิงวัตถุได้ ภาษาทั้งสองแต่เดิมทำให้เกิดผลโดยใช้ตัวแปลภาษาแบบแปลรหัสตัวต่อรหัส นั่นคือรหัสต้นฉบับของภาษาดังกล่าวจะถูกแปลเป็นรหัสภาษาซีก่อน จากนั้นจึงแปลด้วยคอมไพเลอร์อีกต่อหนึ่ง

ภาษาซีพลัสพลัสประดิษฐ์ขึ้นโดยเบียเนอ สเตราสตร็อบ (Bjarne Stroustrup) ให้เป็นภาษาที่มีการทำงานเชิงวัตถุโดยมีวากยสัมพันธ์คล้ายภาษาซี ภาษาซีพลัสพลัสเพิ่มเติมความรัดกุมต่อชนิดตัวแปร ขอบข่าย และเครื่องมืออื่น ๆ ที่เป็นประโยชน์ในการเขียนโปรแกรมเชิงวัตถุ และอนุญาตให้เขียนโปรแกรมเชิงทั่วไประหว่างแบบ ภาษาซีพลัสพลัสรองรับรหัสส่วนใหญ่ของภาษาซีจนแทบจะครอบคลุมทั้งหมด แต่ก็มีข้อยกเว้นบางประการ (ดูเพิ่มที่ ความเข้ากันได้ระหว่างภาษาซีและภาษาซีพลัสพลัส สำหรับรายการความแตกต่างโดยละเอียด)

ภาษาอ็อบเจกทีฟ-ซีเดิมเป็นเพียง "ชิ้นบาง ๆ" บนภาษาซี และยังคงครอบคลุมภาษาซีอย่างเข้มงวด ซึ่งอนุญาตให้เขียนโปรแกรมเชิงวัตถุโดยใช้กระบวนการต้นฉบับตัวแปรผสมพลวัต/สถิต วากยสัมพันธ์ของภาษาอ็อบเจกทีฟ-ซีมาจากทั้งภาษาซีและภาษาสมอลล์ทอล์ก นั่นคือ

วากยสัมพันธ์ที่เกี่ยวข้องกับการประมวลผลก่อน นิพจน์ การประกาศฟังก์ชัน และ การ เรียก ใช้ ฟังก์ชัน รับ มา จา ก ภาษา ซี ในขณะทีวากยสัมพันธ์สำหรับคุณลักษณะเชิงวัตถุนำมาจากภาษาสมอลล์ทอล์ก

ภาษาดีทำคุณลักษณะหลายอย่างให้ต่างออกไปแต่ยังคงไว้ซึ่งวากยสัมพันธ์ทั่วไปของภาษา ซี ไม่เหมือนภาษาซีพลัสพลัสที่แทบจะเข้ากันได้แบบย้อนหลังกับภาษาซี ภาษาดีละทิ้งคุณลักษณะจำนวนหนึ่งของภาษาซีออกไป เนื่องจากวอลเตอร์ ไบรต์ (Walter Bright) ผู้ออกแบบภาษาดี พิจารณาว่าไม่มีความจำเป็นต้องใช้คุณลักษณะเหล่านั้น รวมทั้งตัวประมวลผลก่อนและไตรอักษร ส่วนขยายบางอย่างของภาษาดีไปยังภาษาซี ทับซ้อนกับส่วนขยายไปยังภาษาซีพลัสพลัส

ภาษาดีมีโบ เป็น ภาษา หนึ่ง ที่ พัฒนา โดย ทีมงาน ที่ เบิลล์ แล็บส์ และในขณะที่ยังคงรักษาวากยสัมพันธ์และลักษณะทั่วไปบางอย่างของภาษาซี ก็ยังมีการเก็บกวาดข้อมูลขยะและภาวะพร้อมกันที่มีพื้นฐานบนกระบวนการสื่อสารแบบลำดับ (communicating sequential processes)

ภาษาไพทอน สืบ ทอ มา จาก ภาษา ซี ใน แนวทาง ที่ ต่าง ออก ไป ในขณะที่วากยสัมพันธ์และความหมายของภาษาไพทอนแตกต่างกับภาษาซีอย่างสิ้นเชิง แต่ เครื่องมือ ทำให้ เกิด ผล ใน ภาษา ไพทอนที่ใช้กันอย่างกว้างขวางที่สุดคือซีไพทอน ซึ่งเป็นโปรแกรมภาษาซีแบบโอเพนซอร์ซ สิ่งนี้ช่วยให้ผู้ใช้สามารถเขียนภาษาซีเป็นส่วนขยายของภาษาไพทอน หรือ ฟังก์ชัน ภาษา ไพทอน ลง ใน โปรแกรม ภาษา ซี ความสัมพันธ์อย่างใกล้ชิดนี้เป็นปัจจัยหนึ่งที่นำไปสู่ความสำเร็จของภาษาไพทอนในฐานะภาษาพลวัตเพื่อการใช้งานทั่วไป

ภาษาเพิร์ลเป็นอีกตัวอย่างหนึ่งของภาษาโปรแกรมที่มีต้นกำเนิดจากภาษาซี โครงสร้างโดยรวมทั้งหมดของภาษาเพิร์ลมาจากภาษาซีอย่างมาก เครื่องมือ ทำให้ เกิด ผล ของ ภาษา เพิร์ล มา ตร ฐาน เขียน ขึ้น ด้วย ภาษา ซี และรองรับส่วนขยายที่เขียนในภาษาซีด้วย

## 2.4 การใช้ if else

### 2.4.1 คำสั่ง if

คำสั่ง If เป็นคำสั่งที่เป็นพื้นฐานที่สุดสำหรับควบคุมการทำงานในภาษา C++ คำสั่ง If ถูกใช้เพื่อควบคุมโปรแกรมกับเงื่อนไขที่กำหนด โค้ดในบล็อกของ คำสั่ง If จะ ทำงาน ถ้า เงื่อนไข ตรง หรือ เป็นจริง เรามักจะใช้ คำสั่ง If

ในกรณีที่โปรแกรมนั้นต้องทำงานภายใต้เงื่อนไขบางอย่าง นี่เป็นรูปแบบของการใช้งานคำสั่ง if ในภาษา C++

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

ในรูปแบบการทำงานของคำสั่ง if นั้นจะทำการตรวจสอบเงื่อนไขที่สร้างจาก expression ภายในบล็อกของคำสั่งนั้นจะล้อมรอบด้วยวงเล็บปีกกา {...} ภายในบล็อกนั้นสามารถประกอบไปด้วยตั้งแต่หนึ่งถึงหลายคำสั่ง นี่เป็นตัวอย่างการใช้งานคำสั่ง if ในภาษา C++

```
#include <iostream>

using namespace std;

int main()
{
    int n = 10;
    if (n == 10)
    {
        cout << "n is 10";
    }

    return 0;
}
```

ในตัวอย่างนี้ เราได้ใช้คำสั่ง if เพื่อตรวจสอบว่าตัวแปร n มีค่าเท่ากับ 10 หรือไม่ ถ้าตรงกับเงื่อนไขหรือเงื่อนไขเป็นจริง โปรแกรมจะทำงานในบล็อกของ if คือ cout << "n is 10" ซึ่งเป็นการแสดงผลข้อความว่า "n มีค่าเป็น 10"

#### 2.4.2 คำสั่ง If else

คำสั่ง If else นั้นคล้ายกับคำสั่ง if คำสั่ง if else นั้นจะใช้สำหรับการสร้างเงื่อนไขแบบหลายทางเลือก และมันจะต้องเริ่มต้นด้วยคำสั่ง if เสมอ และนอกจากนี้ยังมีคำสั่ง else clause เพื่อทำในเงื่อนไขที่นอกเหนือจากเงื่อนไขอื่นทั้งหมด มาดูตัวอย่าง

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
    int n = -1;
    if (n < 0)
    {
        cout << "Negative number.";
    }
    else if (n > 0)
    {
        cout << "Positive number.";
    }
    else
    {
        cout << "Zero number";
    }
    return 0;
}
```

ในตัวอย่าง คำสั่ง if สามารถมีเงื่อนไขได้หลายอัน โดยการใช้ else if สำหรับสร้างเงื่อนไขที่สองเป็นต้นไป และในเงื่อนไขสุดท้ายคือ else ซึ่งมันจะทำงานเมื่อไม่ตรงกับเงื่อนไขใดๆ ก่อนหน้า ในโค้ดเรามีตัวแปร n ซึ่งมีข้อมูลเป็นแบบ Integer



โปรแกรมของเรานั้นจะตรวจสอบว่า  $n$  เป็นจำนวนเต็มบวก เต็มลบ หรือศูนย์มากไปกว่านั้น เราสามารถมีได้หลายเงื่อนไขย่อย ๆ โดยใช้ตัวดำเนินการตรรกะสำหรับการเชื่อม Expression เข้าด้วยกัน

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int a = 12;
```

```
    int b = 5;
```

```
    if (a > 10 && b % 2 == 0)
```

```
    {
```

```
        cout << "a is greater than 10 and b is even number.";
```

```
    }
```

```
    else
```

```
    {
```

```
        cout << "Other condition";
```

```
    return 0;
```

ในตัวอย่างนั้นใช้หลายเงื่อนไขและผลลัพธ์ของโปรแกรมจะเป็น "Other condition" เพราะว่า  $a$  นั้นมากกว่า 10 แต่  $b$  นั้นไม่เป็นจำนวนคู่ คุณสามารถเพิ่มเงื่อนไขได้อีกตามที่คุณต้องการ

ตัวอย่างการใช้งานคำสั่งเลือกเงื่อนไขในภาษา C++

หลังจากคุณได้รู้จักกับคำสั่งควบคุมแบบเลือกเงื่อนไขในแบบต่างๆ แล้ว เพื่อให้คุณเข้าใจมากขึ้นต่อไปจะเป็นตัวอย่างของโปรแกรมคำนวณเกรดจากคะแนน โดยเราจะใช้คำสั่งทั้งหมดที่คุณเพิ่งได้เรียนรู้มา เช่น คำสั่ง if else-if และ else

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
{
    int score = 0;
    string name;
    char grade;

    cout << "Enter you name: ";
    cin >> name;
    cout << "Enter your score (0 - 100): ";
    cin >> score;

    if (score >= 0 && score <= 100)
    {

        if (score >= 80)
        {
            grade = 'S';
        }
        else if (score >= 70)
        {
            grade = 'A';
        }
        else if (score >= 60)
        {
            grade = 'B';
        }
        else
        {
            grade = 'C';
        }
    }
}
```

```

    cout << "Hello " << name << endl;
    cout << "You got grade " << grade << endl;

}
else
{
    cout << name << ", you entered an invalid score." << endl;
    cout << "Please try again." << endl;
}
return 0;
}
if (score >= 0 && score <= 100)
{ ...
}
else
{
    cout << name << ", you entered an invalid score." << endl;
    cout << "Please try again." << endl;
}

```

ในกรณีที่ทำงานของโปรแกรม เราได้ทำการตรวจสอบคะแนนด้วยคำสั่ง if ถ้าหากคะแนนนั้นอยู่ระหว่าง 0 ถึง 100 เราจะทำการคำนวณ และถ้าหากไม่ใช่โปรแกรมจะทำงานในบล็อกของคำสั่ง else และบอกว่ากรอกคะแนนไม่ถูกต้อง

```

if (score >= 80)
{
    grade = 'S';
}
else if (score >= 70)

```

```

{
    grade = 'A';
}
else if (score >= 60)
{
    grade = 'B';
}
else
{
    grade = 'C';
}

```

การใส่เงื่อนไขคำสั่ง if นั้นสามารถซ้อนกันไปได้ ดังนั้นในคำสั่ง if ภายในจะทำงานก็ต่อเมื่อเงื่อนไขของ if บล็อกนอกเป็นจริง คือคะแนนที่กรอกเข้ามานั้นอยู่ระหว่าง 0 ถึง 100 เราได้สร้างเงื่อนไขแบบหลายทางเลือกสำหรับตรวจสอบเกรดด้วยคำสั่ง else if ซึ่งก็มีสามเงื่อนไข และในคำสั่ง else นั้นจะทำงานในกรณีที่โปรแกรมไม่ตรงกับทั้งสามเงื่อนไขก่อนหน้าเลย

Enter you name: Mateo

Enter your score (0 - 100): 85

Hello Mateo

You got grade S

นี่เป็นผลลัพธ์การทำงานของโปรแกรม จะเห็นว่าคะแนน 85 ที่กรอกเข้ามานั้นทำให้เงื่อนไข if ครั้งแรกเป็นจริง และทำงานภายในคำสั่ง If ภายในก็ตรงเป็นเงื่อนไขแรก ทำให้ได้เกรด S

Enter you name: Mateo

Enter your score (0 - 100): 105

Mateo, you entered an invalid score.

Please try again.

หลังจากนั้นเราได้รันโปรแกรมอีกครั้ง และกรอกคะแนนเป็น 105 ทำให้เงื่อนไขในคำสั่ง If ด้านนอกไม่เป็นจริง และโปรแกรมทำงานในบล็อกของคำสั่ง else

คำสั่ง Switch case

คำสั่ง switch-case นั้นคล้ายกับ คำสั่ง If-else เป้าหมายของมันเพื่อตรวจสอบกับค่าคงที่นี้เป็นตัวอย่างการใช้คำสั่ง switch

```
#include <iostream>
using namespace std;
int main()
{
    int n = 1;
    switch (n)
    {
        case 1:
            cout << "n is 1";
            break;
        case 2:
            cout << "n is 2";
            break;
        default:
            cout << "Unknown n";
    }
    return 0;
}
```

switch expression สามารถมีได้แค่หนึ่งค่า เพื่อประเมินคำสั่ง case เป็นคำสั่งเงื่อนไขเพื่อเปรียบเทียบค่า ในตัวอย่าง case 1: จะทำงานเมื่อ n มีค่าเท่ากับ 1 หลังจากคำสั่งด้านล่างเราต้องใส่คำสั่ง break เพื่อหยุดสำหรับแต่ละ

case

ไม่เช่นนั้น โปรแกรมจะทำงานไปจนกว่าจะพบคำสั่ง break หรือสิ้นสุดบล็อกคำสั่งของ switch และคำสั่ง default นั้นเป็นทางเลือกเมื่อโปรแกรมไม่ตรงกับเงื่อนไขใดๆ ก่อนหน้าเช่นเดียวกับคำสั่ง else ในตัวอย่าง มันสามารถถูกเขียนโดยการใช้คำสั่ง if-else ได้ดังด้านล่างนี้

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
    int n = 1;
    if (n == 1)
    {
        cout << "n is 1";
    }
    else if (n == 2)
    {
        cout << "n is 2";
    }
    else
    {
        cout << "Unknown n";
    }
    return 0;
}
```

ในตัวอย่าง คุณจะเห็นว่าการใช้งานคำสั่ง switch นั้นสามารถที่จะแทนที่ด้วยคำสั่ง if ได้ แต่อย่างก็ตามคำสั่ง switch นั้นยังมักจะถูกใช้กับการเปรียบเทียบที่ง่ายและรวดเร็ว ดังนั้น switch นั้นจึงใช้ได้เพียงกับข้อมูลประเภท Integer หรือ Char เท่านั้น ในขณะที่ if นั้นสามารถใช้กับข้อมูลทุกแบบ

คำสั่ง while loop

ลูปที่ง่ายและพื้นฐานที่สุดในภาษา C++ นั่นคือ while loop เราใช้คำสั่งลูปสำหรับวนซ้ำให้โปรแกรมสามารถทำงานเดิมซ้ำๆ ได้ ซึ่งมีรูปแบบการใช้งานคือ

```
while (expression)
```

```
{
    statements
}
```

คำสั่ง while loop ใช้เพื่อทำสิ่งใดของโปรแกรมในขณะที่ Expression เป็นจริง true และมันจะสิ้นสุดการทำงานเมื่อ Expression เป็นเท็จ false และออกจาก while loop และทำคำสั่งอื่นต่อไป

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
    int n = 1;
    while (n <= 10)
    {
        cout << n << " ";
        n++;
    }
    cout << " end loop";
    return 0;
}
```

ในตัวอย่าง โปรแกรมจะนับจาก 1 ถึง 10 เราได้ประกาศตัวแปร n และกำหนดค่าให้เป็น 1 ก่อนที่มันจะเข้าไปในทำงานใน while loop

และคำสั่ง while loop จะทำการตรวจสอบ Expression และเข้าสู่ถ้าเงื่อนไขยังคงเป็นจริง และแสดงค่า n ออกทางจอภาพและเพิ่มค่า n ขึ้น 1 จนกว่า n จะเพิ่มไปถึง 10 ซึ่งจะทำให้ Expression เป็นเท็จ และโปรแกรมจะออกจาก loop และทำสิ่งอื่นต่อไป

และนี่เป็นผลลัพธ์เมื่อรันโปรแกรม

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

คำสั่ง do while loop

ลูปที่คล้ายกับ while loop คือ do while ลูป มันมีรูปแบบดังนี้

do

{

statements

}

while (condition);

มันทำางานเหมือน while loop ยกเว้นในการเปรียบเทียบเงื่อนไขจะทำตอนท้ายหลังจากสิ้นสุดคำสั่งในลูป นั่นหมายความว่า do while loop จะต้องทำางานอย่างน้อยหนึ่งรอบแน่นอน มันมักจะใช้กับโปรแกรมที่จำเป็นต้องรับค่าจากผู้ใช้ก่อนที่จะทำอย่างอื่นต่อไป มาดูตัวอย่างที่ง่าย ๆ

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    char ch;
```

```
    do
```

```
    {
```

```
        cout << "Enter 'n' to exit loop: ";
```

```
        cin >> ch;
```

```
    }
```

```
    while (ch != 'n');
```



```

    cout << "Exit from the loop.";
    return 0;
}

```

โปรแกรมข้างบนต้องการรับค่าจากผู้ใช้ expression ของมันต้องการตัวอักษร 'n' เพื่อออกจากลูป นี่คือผลลัพธ์ของโปรแกรมเมื่อได้ทดสอบ คุณสามารถลองดูได้เช่นกัน

Enter 'n' to exit loop: a

Enter 'n' to exit loop: b

Enter 'n' to exit loop: c

Enter 'n' to exit loop: n

Exit from the loop.

### 2.4.3 คำสั่ง for loop

for loop เป็นลูปที่มีการวนรอบเป็นจำนวนที่แน่นอน รูปแบบของมันคือ

```

for (initialize; condition; increase)
{
    statements
}

```

for loop เป็นลูปที่สามารถวนรอบตามตัวเลขที่กำหนดได้ มันทำงานเหมือน while loop มันจะวนซ้ำจนกว่า expression จะเป็นเท็จ นอกจากนั้น เรายังสามารถประกาศตัวแปรเริ่มต้น สร้าง expression เพิ่มและลดค่าก่อนที่ลูปจะเริ่ม ตัวอย่างการนับตัวเลขโดยใช้ for loop

```

#include <iostream>

using namespace std;

int main()
{

```

```

for (int n = 1; n <= 10; n++)
{
    cout << n << ",";
    cout << " end loop";
    return 0;
}

```

และนี่เป็นผลลัพธ์เมื่อเรารันโปรแกรม ซึ่งมันเป็นโปรแกรมเดียวกันกับตัวอย่างของ while loop ก่อนหน้า

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

#### 2.3.4 คำสั่ง break

คำสั่ง break ใช้สำหรับเพื่อจบการทำงานของลูปในทันที และมันไม่สนใจว่า expression จะ เป็น จริง หรือ ไม่ เมื่อ พบ กับ คำ สั่ง break โปรแกรมจะออกนอกลูปและทำงานบรรทัดต่อไปหลังจากลูปทันที เรามักจะใช้คำสั่งนี้สำหรับการออกจากลูปโดยเงื่อนไขพิเศษ

```
#include <iostream>
```

```
using namespace std;
```

```

int main()
{
    for (int n = 1; n <= 10; n++)
    {
        if (n == 5)
            break;
        cout << n << ",";
    }
    cout << " end loop";
    return 0;
}

```

จากตัวอย่างข้างบน เป็นโปรแกรมในการแสดงตัวเลข 1 ถึง 10 แต่ในตอนนี้ โปรแกรมจะออกจากลูปเมื่อ  $n$  มีค่าเท่ากับ 5 เพราะว่าเราได้สร้างเงื่อนไขเมื่อ  $n$  มีค่าเท่ากับ 5 ให้คำสั่ง break ทำงาน คำสั่ง break สามารถใช้กับลูป เช่น for, while, do while, switch และอื่นๆ และผลลัพธ์การทำงานจะเป็นดังนี้

1,2,3,4, end loop

#### 2.4.4 คำสั่ง continue

ไม่เหมือนคำสั่ง break คำสั่ง continue ถูกใช้เพื่อข้ามการทำงานในรอบปัจจุบัน ซึ่งจะไม่ทำคำสั่งหลังจากมันและไปเริ่มรอบถัดไปในทันที มาดูตัวอย่างการใช้งาน continue ในภาษา C++

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

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

```
    {
```

```
        if (n % 2 == 0)
```

```
            continue;
```

```
        cout << n << " ";
```

```
    }
```

```
    cout << " end loop";
```

```
    return 0;
```

```
}
```

ในตัวอย่าง เราได้เขียนโปรแกรมในการแสดงตัวเลขจาก 1 ถึง 10 เช่นเดิม แต่ โปรแกรม จะ ข้าม ลูป ถ้า  $n$  เป็น ตัว เลข คู่ เรา ได้ ใช้ คำ สั่ง If

ในการสร้างเงื่อนไขว่าถ้าหากเป็นเลขคู่ นั้นให้ทำงานคำสั่ง continue ทำให้คำสั่ง cout ที่อยู่หลังคำสั่ง continue ถูกข้ามการทำงานไป และโปรแกรมแสดงผลเพียงเลขที่ 1, 3, 5, 7, 9, end loop ตัวอย่างการใช้งานคำสั่งวนซ้ำในภาษา C++ หลังจากคุณได้เรียนรู้และทราบการทำงานของคำสั่งวนซ้ำประเภทต่างๆ แล้วต่อไปจะเป็นตัวอย่างการใช้งานคำสั่งต่างๆ รวมกัน เราจะเขียนโปรแกรมช่วยการทอนเงินของผู้ขาย โดยที่จะทอนเหรียญที่มีค่ามากที่สุดก่อน นี่เป็นโค้ดของโปรแกรม

```
#include <iostream>

using namespace std;

int main()
{
    const int SIZE = 8;
    int coin[SIZE] = {1000, 500, 100, 50, 20, 10, 5, 1};
    int count[SIZE] = {0, 0, 0, 0, 0, 0, 0, 0};

    int money;
    cout << "Enter money: ";
    cin >> money;

    for (int i = 0; i < SIZE; i++)
    {
        count[i] = money / coin[i];
        money %= coin[i];
    }

    cout << "Your change" << endl;

    for (int i = 0; i < SIZE; i++)
    {
        if (count[i] == 0)
        {
```

```

        continue;
    }

    cout << coin[i] << " => " << count[i] << endl;
}

return 0;
}

```

ในตัวอย่าง เป็นโปรแกรมในการทอนเงินจากการซื้อของ โดยโปรแกรมจะให้กรอกจำนวนเงินที่ต้องการจะทอนและบอกว่าต้องทอนเงินเหรียญและธนบัตร ข น ำ ค ต ำ ง ๆ อ ย ำ ง ล ะ เ ท ำ ไ ร

ในส่วนแรกของโปรแกรมเราได้กำหนดประเภทของเหรียญและธนบัตรในตัวแปร อาร์เรย์ coin และ ตัวแปรอาร์เรย์ count สำหรับนับเหรียญและธนบัตรแต่ละประเภท และโปรแกรมของเราเริ่มทำงานจากการรับจำนวนเงินทอนเก็บในตัวแปร money

```

for (int i = 0; i < SIZE; i++)
{
    count[i] = money / coin[i];
    money %= coin[i];
}

```

ต่อมาเราได้ใช้คำสั่ง for loop ในการวนรอบอาร์เรย์ coin โดยวนจากประเภทที่มากที่สุดก่อน และเพื่อหาจำนวนเงินที่ต้องทอนของแต่ละประเภทว่าต้องทอนเท่าไร โดยจะนำจำนวนเงินทั้งหมดหารด้วยประเภทของเงินและเก็บไว้ในตัวแปร count หลังจากนั้นเราลบจำนวนเงินนั้นออกไปโดยการใช้ดำเนินการ mod

```

for (int i = 0; i < SIZE; i++)
{
    if (count[i] == 0)
    {
        continue;
    }
}

```

```

}

cout << coin[i] << " => " << count[i] << endl;

}

```

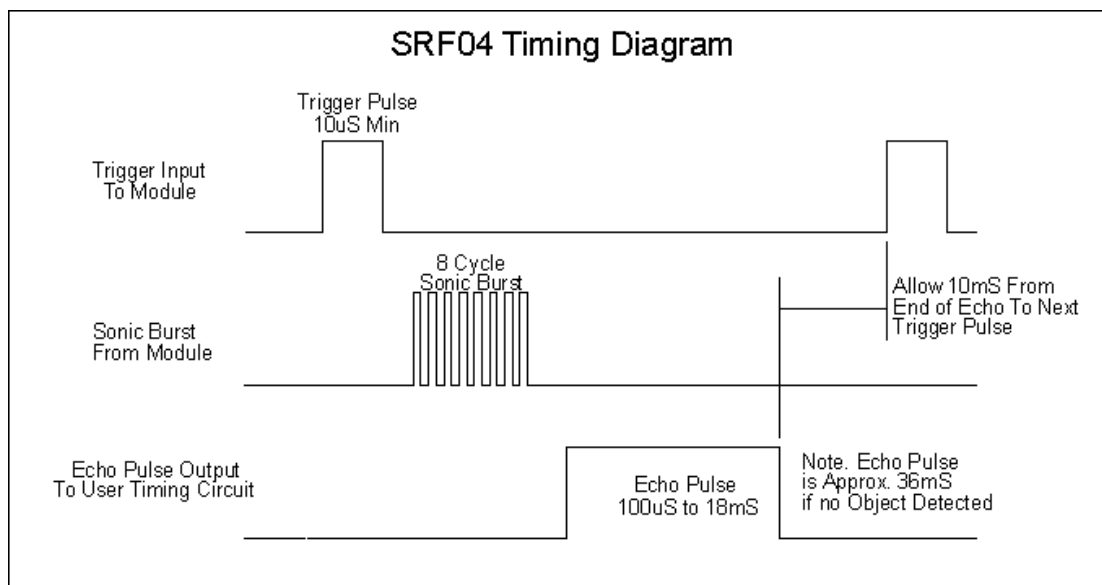
หลังจากการหาจำนวนเงินแต่ละประเภทที่ต้องทอนเสร็จสิ้นแล้ว เราใช้คำสั่ง for loop ในการวนเพื่อแสดงผลว่าจะต้องทอนเงินแต่ละประเภทเท่าไร และเราใช้คำสั่ง continue ในการข้ามการแสดงผลประเภทของเงินที่ไม่มีการทอนไป

## 2.5 การคำนวณระยะทาง

เซ็นเซอร์หลายรุ่น ใช้วิธีนี้ในการติดต่อสื่อสารกับไมโครคอนโทรลเลอร์ ซึ่งในแต่ละรุ่นจะใช้จำนวนสายไม่เท่ากัน ในบางรุ่นจะใช้สาย 2 เส้น คือ Trig สำหรับส่งสัญญาณ และ Echo สำหรับรับสัญญาณกลับมา และในบางรุ่นจะใช้เส้นเดียว คือทั้ง Trig และ Echo อยู่เส้นเดียวกันเลย และใช้วิธีแบ่งเวลารับ - ส่งข้อมูล (หลักการเหมือน 1-wire bus)

ในการสื่อสารแบบทริกสัญญาณ เริ่มต้นจะต้องให้สัญญาณขา Trig มีสถานะทางลอจิกเป็น LOW เสียก่อน จากนั้นจึงเริ่มทริกสัญญาณ โดยให้ขา Trig มีสถานะเป็น HIGH ค้างไว้อย่างน้อย 10uS แล้วจึงปรับสถานะเป็น LOW

จากนั้น ที่ขา Echo ให้เตรียมรับสัญญาณทริก HIGH กลับมา เมื่อมีการส่งสัญญาณ HIGH กลับมา ให้เริ่มนับเวลาที่สัญญาณเป็น HIGH และเมื่อสัญญาณขา Echo กลับเป็น LOW ให้สิ้นสุดการนับเวลา แล้วจึงนำค่าเวลาที่นับได้ ไปคำนวณอีกที ซึ่งในการคำนวณนั้น จะขึ้นอยู่กับรุ่น ในบางรุ่นสามารถใช้ค่าอัตราเร็วเสียงมาคำนวณได้เลย แต่ในบางรุ่น ต้องใช้สูตรคำนวณเฉพาะ



รูปที่ 2.2 Timing Diagram ของเซ็นเซอร์วัดระยะด้วยคลื่นอัลตราโซนิก HCSR04

([https://cz.lnwfile.com/\\_/cz/\\_raw/vn/gy/gk.gif](https://cz.lnwfile.com/_/cz/_raw/vn/gy/gk.gif))

โดยส่วนใหญ่แล้ว จะใช้ค่าอัตราเร็วเสียงหาร 2 ได้เลย ที่นำมาหาร 2 เพราะระยะทางที่วัดได้ คือระยะทางทั้งขาไป และขากลับ เมื่อนำมาหาร 2 เราจะได้แค่ระยะห่างตอนขาไป หรือขากลับนั่นเอง

ตัวอย่างของโค้ดที่ใช้กับ Arduino มีดังนี้

```
/*
 * Codeing By IOXhop : www.ioxhop.com
 */

#define Trig_PIN 12 // Pin connect to Trig pin
#define Echo_PIN 13 // Pin connect to Echo pin

void setup() {
  pinMode(Trig_PIN, OUTPUT);
  pinMode(Echo_PIN, INPUT);
  Serial.begin(9600);
}

void loop() {
  digitalWrite(Trig_PIN, LOW);
```

```

delayMicroseconds(5);

digitalWrite(Trig_PIN, HIGH);

delayMicroseconds(10);

digitalWrite(Trig_PIN, LOW);

unsigned int PulseWidth = pulseIn(Echo_PIN, HIGH);

unsigned int distance = PulseWidth * ???;

Serial.print("Distance is ");

Serial.print(distance);

Serial.println(" cm.");

delay(100);

}

```

จากโค้ดด้านบน หากโมดูลใช้อัตราเร็วของเสียงในการคำนวณ ท่านสามารถแทนที่ ??? ด้วย 0.0173681 ได้เลย โดยค่านี้นี้หามาได้จากสูตรหาอัตราเร็วของเสียงที่กำหนดให้อุณหภูมิ ณ ขณะนี้เป็น 27 องศาเซลเซียส

เมื่อ

อัตราเร็วของเสียงในอากาศ =  $331 + (0.606 * \text{อุณหภูมิในหน่วยองศาเซลเซียส})$  m/s

อัตราเร็วของเสียงในอากาศ =  $331 + (0.606 * 27)$  m/s

อัตราเร็วของเสียงในอากาศ = 347.362 m/s

จากนั้นให้พิจารณาว่า ฟังก์ชัน pulseIn(pin, value) จะส่งค่ากลับมาในหน่วยไมโครวินาที (uS) เราจึงต้องแปลงให้ค่า 347.362 m/s ให้เป็นหน่วย m/uS โดยนำไปคูณด้วย  $10^{-6}$  ต่อมาเราต้องการให้ค่าออกมาในหน่วยเซนติเมตร (cm) ให้นำไปคูณ  $10^2$  (เพราะ 1 เมตร = 100 เซนติเมตร) จึงจะได้ค่าออกมาในหน่วย เซนติเมตร ต่อ ไมโครวินาที (cm/uS)

อัตราเร็วของเสียงในอากาศ =  $347.362 * 10^{-6} * 10^2$  cm/uS

อัตราเร็วของเสียงในอากาศ =  $347.362 * 10^{(-6 + 2)}$  cm/uS

อัตราเร็วของเสียงในอากาศ = 0.0347362 cm/uS

เนื่องจากสัญญาณเดินทางทั้งไป และกลับ จึงต้องนำค่ามาหารด้วย 2 จึงได้ 0.0173681 นั่นเอง



ระยะห่างระหว่างวัตถุกับหน้าเซ็นเซอร์ = เวลาที่ได้จากขา Echo ในหน่วยไมโครวินาที \* 0.0173681  
cm

การสร้างฟังก์ชัน pulseIn()

หัวใจสำคัญของการสื่อสารแบบทริกสัญญาณ คือการนับเวลาที่สัญญาณเป็น HIGH ในไมโครคอนโทรลเลอร์ตระกูลอื่น ๆ หรือไม่ได้ใช้ Arduino IDE จะไม่มีฟังก์ชันนี้ให้เราสามารถฟังก์ชัน pulseIn() ได้เอง โดยสามารถเลือกใช้ทั้ง Timer หรือใช้ฟังก์ชันเกี่ยวกับ delay ก็ได้ ตัวอย่างโค้ดด้านล่างนี้ เป็นโค้ดที่แสดงการสร้างฟังก์ชัน pulseIn() ขึ้นมาใช้งานเอง โดยตัวอย่างนี้ยังเป็นของ Arduino อยู่

unsigned

long

\_\_pulseIn(int

pin, int

level) {

while(digitalRead(pin) != level) ;

unsigned long time\_s = 0;

while(digitalRead(pin) == level) {

delayMicroseconds(1);

time\_s++;

}

return time\_s;

}

และในโค้ดตัวอย่างด้านล่างนี้ เป็นโค้ดที่ใช้กับ MPLABXpress Evaluation Board

โดยเขียนโค้ดบน MPLAB Xpress Cloud-based IDE

/\*

\* File: main.c

\* Author: max30012540@hotmail.com

\*

\* Created on 3/18/2017 8:09:13 AM UTC

\* "Created in MPLAB Xpress"

\* Full code you can see :

<https://mplabxpress.microchip.com/mplabcloud/example/details/306>

\*/

```
#include <xc.h>
```

```
#include "mcc.h"
```

```
#include <htc.h>
```

```
#define Trig_Pin LATCbits.LATC5
```

```
#define Echo_Pin PORTCbits.RC6
```

```
unsigned long pulseIn() {
```

```
while(Echo_Pin == 0) ;
```

```
unsigned long time_t = 0;
```

```
while(Echo_Pin == 1) {
```

```
__delay_us(1);
```

```
time_t++;
```

```
}
```

```
return time_t;
```

```
}
```

```
void main(void) {
```

```
SYSTEM_Initialize();
```

```
TRISA = 0xF0;
```

```
TRISCbits.TRISC5 = 0; // Set RC5 to OUTPUT
```

```
TRISCbits.TRISC6 = 1; // Set RC6 to INPUT
```

```
LATA = 0x01;
```

```
while(1) {
```

```
Trig_Pin = 0;
```

```
__delay_us(5);
```

```
Trig_Pin = 1;
```

```
__delay_us(10);
```

```
Trig_Pin = 0;
```

```

unsigned long PulseWidth = pulseIn();

unsigned int distance = PulseWidth * 0.0173681;

distance = distance < 5 ? 0 : (distance > 8 ? 3 : distance - 5);

LATA = 0x01<<(distance);

__delay_ms(100);
}

return;
}

```

## 2.6 วิธีการติดตั้ง Driver Arduino

### 2.6.1 ดาวน์โหลด

ให้ไปที่ <http://arduino.cc/en/Main/Software> และกด click ที่ download เพื่อเข้าสู่ the download page.



รูปที่ 2.3 Download The Arduino Software ที่มา([https://o.lnwfile.com/\\_/o/\\_raw/sv/ob/1e.jpg](https://o.lnwfile.com/_/o/_raw/sv/ob/1e.jpg))

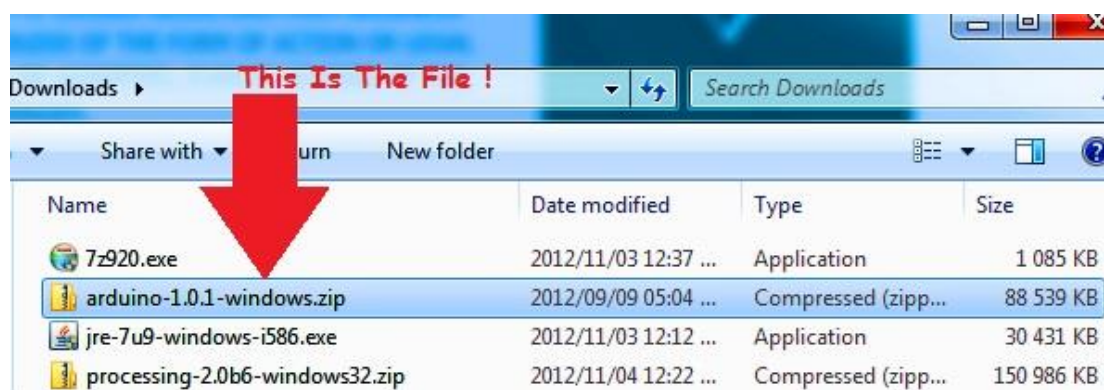
ที่หน้า download, ให้ Click ไปที่ Windows link เพื่อที่จะ download Arduino software สำหรับระบบปฏิบัติการ Windows ตามรูปข้างล่าง



รูปที่ 2.4 Downlad Driver Arduino 1.0.1 ที่มา([https://o.lnwfile.com/\\_/o/\\_raw/l4/m5/lv.jpg](https://o.lnwfile.com/_/o/_raw/l4/m5/lv.jpg))

## 2.6.2 ติดตั้ง

หลังจาก download แล้ว ให้เลือกที่อยู่ของ downloaded file บนระบบ system และทำการ extract the folder จาก zipped file วางลงบนตำแหน่งที่เหมาะสม



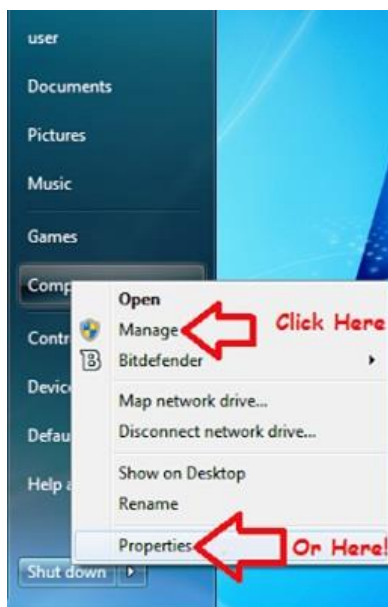
รูปที่ 2.5 Driver Arduino Zip ที่มา([https://o.lnwfile.com/\\_/o/\\_raw/4q/fu/ps.jpg](https://o.lnwfile.com/_/o/_raw/4q/fu/ps.jpg))

### 2.6.2.1 ทำการ Plug the Arduino Board ลงบนเครื่อง PC

ทำการ Plug the Arduino board ลงบนเครื่อง PC. Windows จะพยายาม install drivers แต่สุดท้ายก็จะไม่สำเร็จ

### 2.6.2.2 Start the Windows Device Manager

ให้ Click ไปที่ปุ่ม Start menu คลิกขวาที่ My Computer บน Startmenu แล้ว click Properties or Manage จาก pop-up menu เพื่อทำการเปิด Device Manager



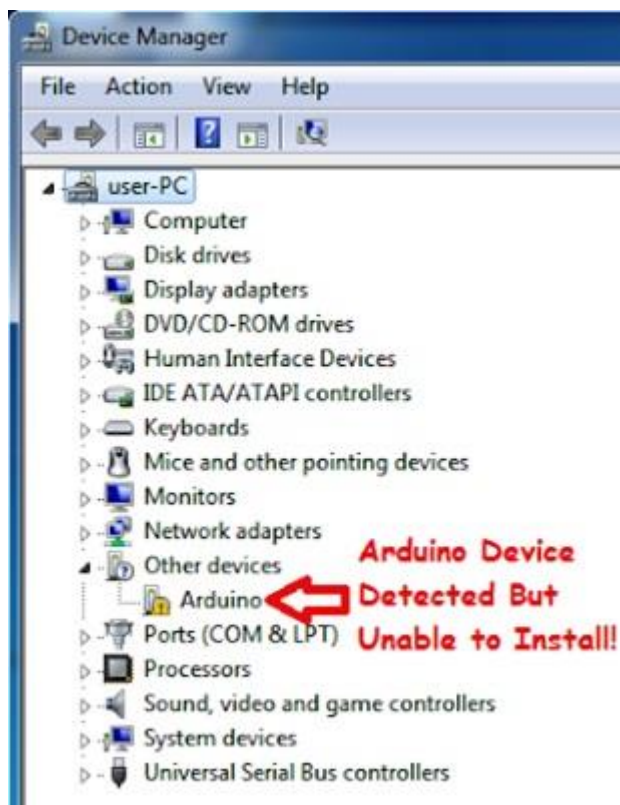
รูปที่ 2.6 Manage Computer ที่มา([https://o.lnwfile.com/\\_/o/\\_raw/jh/m7/e0.jpg](https://o.lnwfile.com/_/o/_raw/jh/m7/e0.jpg))

Click ไปที่ Device Manager link เพื่อ Start device manager:



รูปที่ 2.7 System ที่มา([https://o.lnwfile.com/\\_/o/\\_raw/e4/xa/vi.jpg](https://o.lnwfile.com/_/o/_raw/e4/xa/vi.jpg))

Device Manager จะ เปิด และ แสดง Arduino Device ที่ เรา ทำ การ Connect ใ ไว้  
(ขึ้นอยู่กับว่าใช้ board ชนิดไหน ชื่อก็จะแสดงขึ้นมาให้เห็น

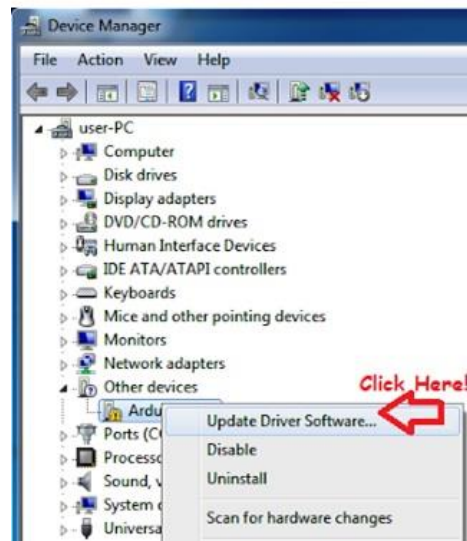


รูปที่ 2.8 Device Manager 1 ที่มา([https://o.lnwfile.com/\\_/o/\\_raw/3v/g9/dw.jpg](https://o.lnwfile.com/_/o/_raw/3v/g9/dw.jpg))

ที่นี้เราจะเจอ เครื่องหมายตกใจ ขึ้นสีเหลืองๆ ซึ่งแสดงว่า อุปกรณ์ Arduino นั้น ไม่สามารถทำการ Install มาถึงขั้นนี้ก็ต้อง

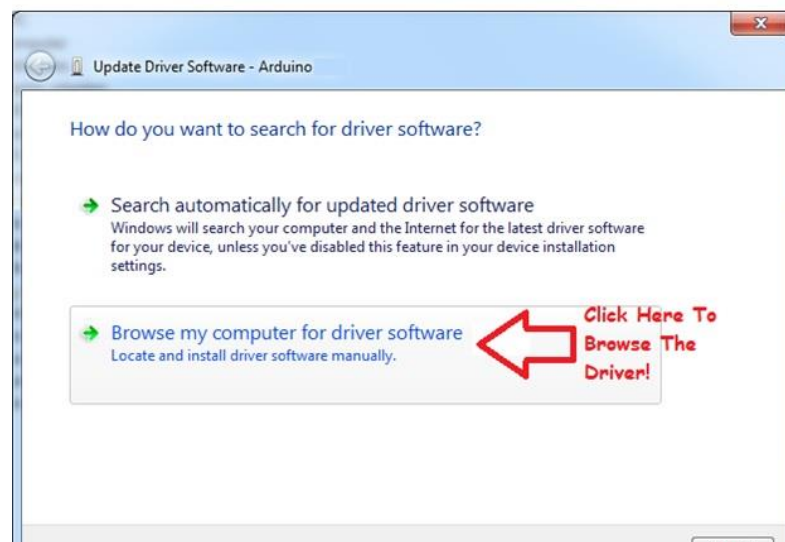
### 2.6.2.3 Installing the Device Driver

ในหน้าต่างของ Device Manager, ให้คลิกขวาไปที่ Arduino board แล้วก็ click Update Driver Software, บน pop-up menu



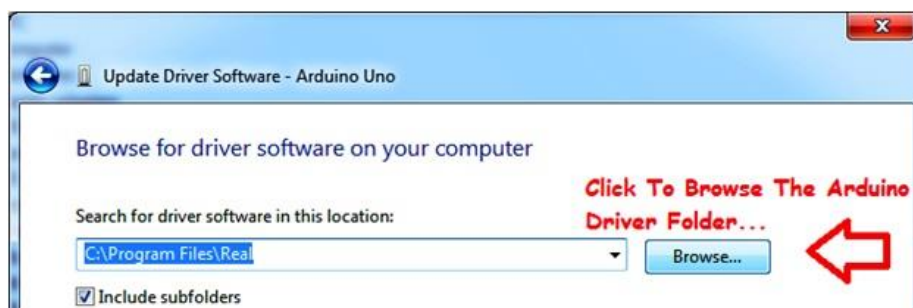
รูปที่ 2.9 Device Manager 2 ที่มา([https://o.lnwfile.com/\\_/o/\\_raw/3v/g9/dw.jpg](https://o.lnwfile.com/_/o/_raw/3v/g9/dw.jpg))

ณ ขณะนี้ จะมีกล่องขึ้น pop-up มาโชว์ว่า Update Driver Software ให้คลิกที่ Browse my computer for driver software เพื่อที่จะ install Driver Software Manually



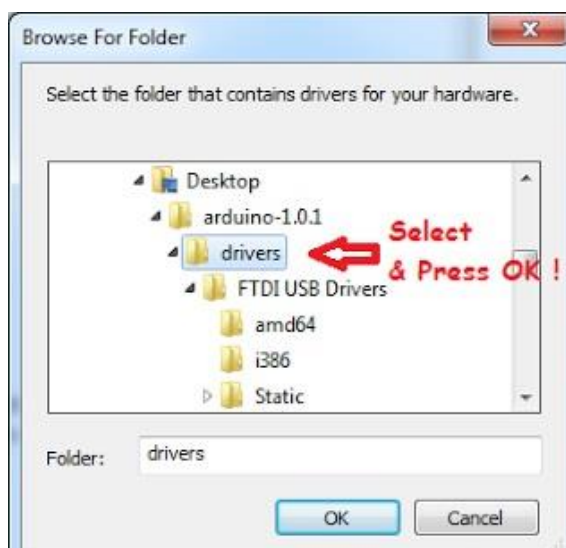
รูปที่ 2.10 Update Driver Software 1 ที่มา([https://o.lnwfile.com/\\_/o/\\_raw/70/gf/23.jpg](https://o.lnwfile.com/_/o/_raw/70/gf/23.jpg))

ต่อไปก็ click the Browse



รูปที่ 2.11 Search For Driver Software ที่มา([https://o.lnwfile.com/\\_/o/\\_raw/b2/1m/hd.jpg](https://o.lnwfile.com/_/o/_raw/b2/1m/hd.jpg))

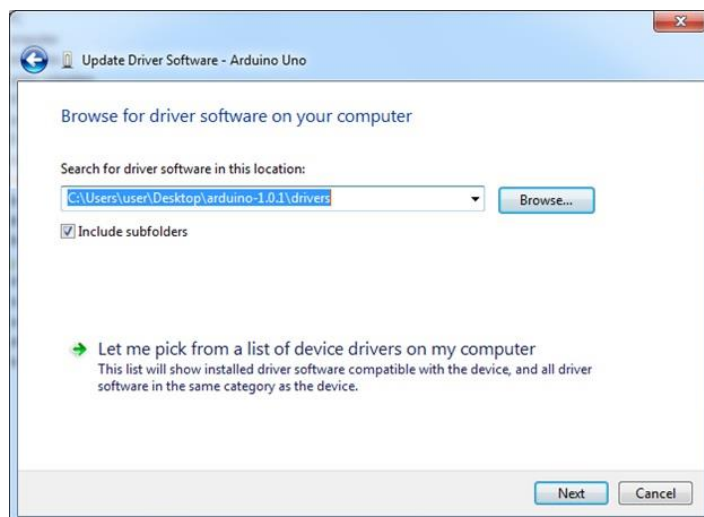
เลือกไปที่ drivers folder ใน Arduino folder ที่ได้ download มา



รูปที่ 2.12 Browse For Folder ที่มา([https://o.lnwfile.com/\\_/o/\\_raw/ys/8v/ju.jpg](https://o.lnwfile.com/_/o/_raw/ys/8v/ju.jpg))

หลังจากเลือกไปที่ driver folder เรียบร้อยแล้ว ให้คลิก Next





รูปที่ 2.13 Update Driver Software 2 ที่มา([https://o.lnwfile.com/\\_/o/\\_raw/fd/tz/bk.jpg](https://o.lnwfile.com/_/o/_raw/fd/tz/bk.jpg))

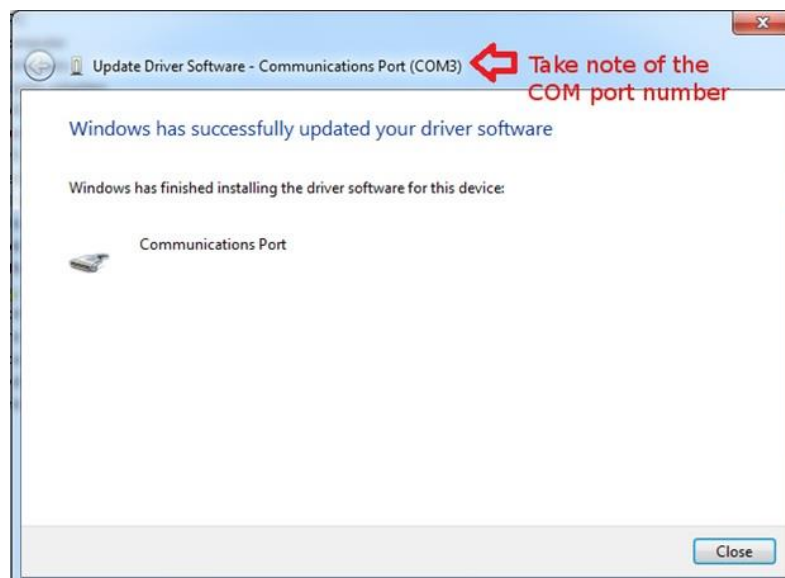
จะมี กล่อง pops up ขึ้นมา, click Install this driver software anyway, to continue the Arduino Board Drivers



รูปที่ 2.14 Windows Security ที่มา([https://o.lnwfile.com/\\_/o/\\_raw/27/pj/fr.jpg](https://o.lnwfile.com/_/o/_raw/27/pj/fr.jpg))

เมื่อทำการ Install Driver เป็นที่เรียบร้อยแล้ว จะมี dialog box ตามรูปข้างล่าง เราต้องทราบ port number ที่

บอร์ด Arduino เราทำการเชื่อมต่ออยู่ในตัวอย่างนี้คือ Port COM3 แต่บอร์ดที่ใช้งานจริงของคุณ อาจจะมาเป็น Port COM อื่นๆ ก็อย่าได้ตกใจครับ

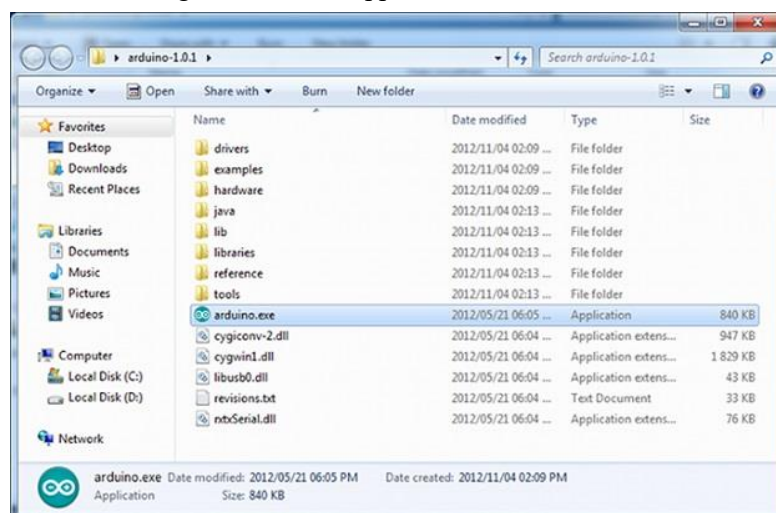


รูปที่ 2.15 Update Driver Software 3 ที่มา([https://o.lnwfile.com/\\_/o/\\_raw/vz/im/8b.jpg](https://o.lnwfile.com/_/o/_raw/vz/im/8b.jpg))

ถ้าคุณเห็นข้อความนี้แสดงว่าคุณได้ทำการ Install Driver เป็นที่เรียบร้อยแล้ว ก็ทำการ "Close"

#### 2.6.2.4 ทำการ Set up Arduino Software หลังจาก Installation แล้ว

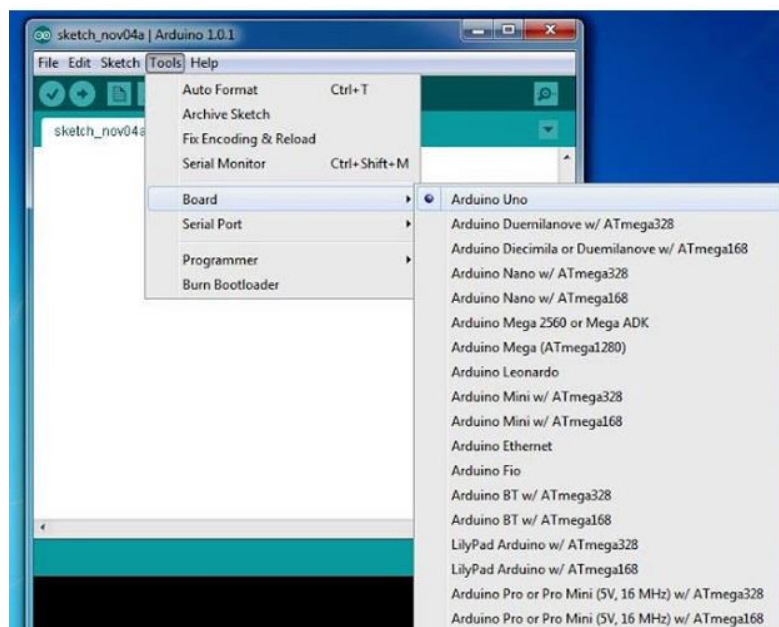
การ set นี้ ทำแค่ครั้งเดียวเป็นอันเสร็จพิธี นอกเสียจากคุณจะต้องการเปลี่ยนรุ่นของ Board หรือ เปลี่ยน port ที่ทำการเชื่อมต่อกับ Board เท่านั้น. เพียงแค่เลือกไปที่ folder ที่คุณได้เก็บ Arduino IDE ไว้ และทำการ start Software ของ Arduino IDE โดยทำการ double-clicking ไปที่ Arduino application.



รูปที่ 2.16 Folder Arduino 1.0.1 ที่มา([https://o.lnwfile.com/\\_/o/\\_raw/pg/3g/50.jpg](https://o.lnwfile.com/_/o/_raw/pg/3g/50.jpg))

Double Click ที่ "Arduino.exe" เพื่อ Start Arduino IDE

จากนั้นเลือกรุ่นของ Arduino board ที่ทำการเชื่อมต่อให้ถูกต้องจากใน list ทำการเปลี่ยน ถ้าจำเป็น



รูปที่ 2.17 Arduino 1.0.1 ที่มา([https://o.lnwfile.com/\\_/o/\\_raw/4f/vl/dn.jpg](https://o.lnwfile.com/_/o/_raw/4f/vl/dn.jpg))

ที่นี้ก็ทำการ check ว่า serial port ที่ต่ออยู่ได้ถูกเลือกอย่างถูกต้อง เลือกไปที่ Computer Serial Port ที่ Arduino ทำการต่อเชื่อมอยู่

### บทที่ 3

#### วิธีการดำเนินโครงการ

ในการดำเนินการสร้างชิ้นงานโครงการ ชุดคำสั่งหมวกเซนเซอร์ตรวจจับสิ่งกีดขวางสำหรับผู้พิการทางสายตา มีขั้นตอนการสร้างในส่วนต่างๆ โดยทางคณะผู้จัดทำได้ร่วมกันวางแผนในการปฏิบัติงาน และจัดแบ่งงานขั้นตอนการดำเนินการสร้างชิ้นงานตามความเหมาะสม

ขั้นตอนในการดำเนินการสร้างชิ้นงาน แบ่งออกเป็นดังนี้

3.1 วางแผนและเตรียมการและเตรียมอุปกรณ์

3.2 ระบบการทำงานของชิ้นงาน

3.3 การสร้างชุดคำสั่ง

#### 3.1 วางแผนการจัดทำชิ้นงานและเตรียมอุปกรณ์

การวางแผนและการเตรียมการ เริ่มเมื่อคณะกรรมการพิจารณาโครงการให้เสนอหัวข้อโครงการในภาคเรียนที่ 1 ทางคณะผู้จัดทำได้เสนอหัวข้อโครงการชุดคำสั่งหมวกเซนเซอร์ตรวจจับสิ่งกีดขวางสำหรับผู้พิการทางสายตา

ซึ่งมีลำดับขั้นตอนต่างๆ ในการดำเนินโครงการดังตารางที่ 3.1

##### 3.1.1 การวางแผนและเตรียมการ

- 1) ศึกษาความเป็นไปได้ของชุดคำสั่งArduino
- 2) หาข้อมูลเพิ่มเติมเกี่ยวกับชุดคำสั่ง
- 3) ศึกษาชุดคำสั่งและประเภทของ Board Arduino แต่ละรุ่น
- 4) หาข้อมูลเพิ่มเติมเกี่ยวกับการใช้ภาษา C++ ควบคุมคำสั่ง
- 5) จัดเตรียมชุดคำสั่งให้พร้อมทำชิ้นงานโครงการ
- 6) นำชุดคำสั่งที่ใช้ทดลองมาลงใช้งานและแก้ไขก่อนจัดชิ้นงานจริง
- 7) เริ่มเขียนชุดคำสั่งลงไปในชิ้นงานโครงการ
- 8) เขียนชุดคำสั่งให้ลงไปใน Board Arduino
- 9) นำ Board Arduino มาติดตั้งที่บนหมวก
- 10) นำ Board Arduino มาเชื่อมกับรางถ่านที่ใช้กระแสไฟ 9.0 V
- 11) ติดตั้ง Switch บริเวณใกล้กับรางถ่านที่ใช้กระแสไฟ 9.0 V
- 12) ตกแต่งส่วนต่างให้เหมาะสมกับชิ้นงาน

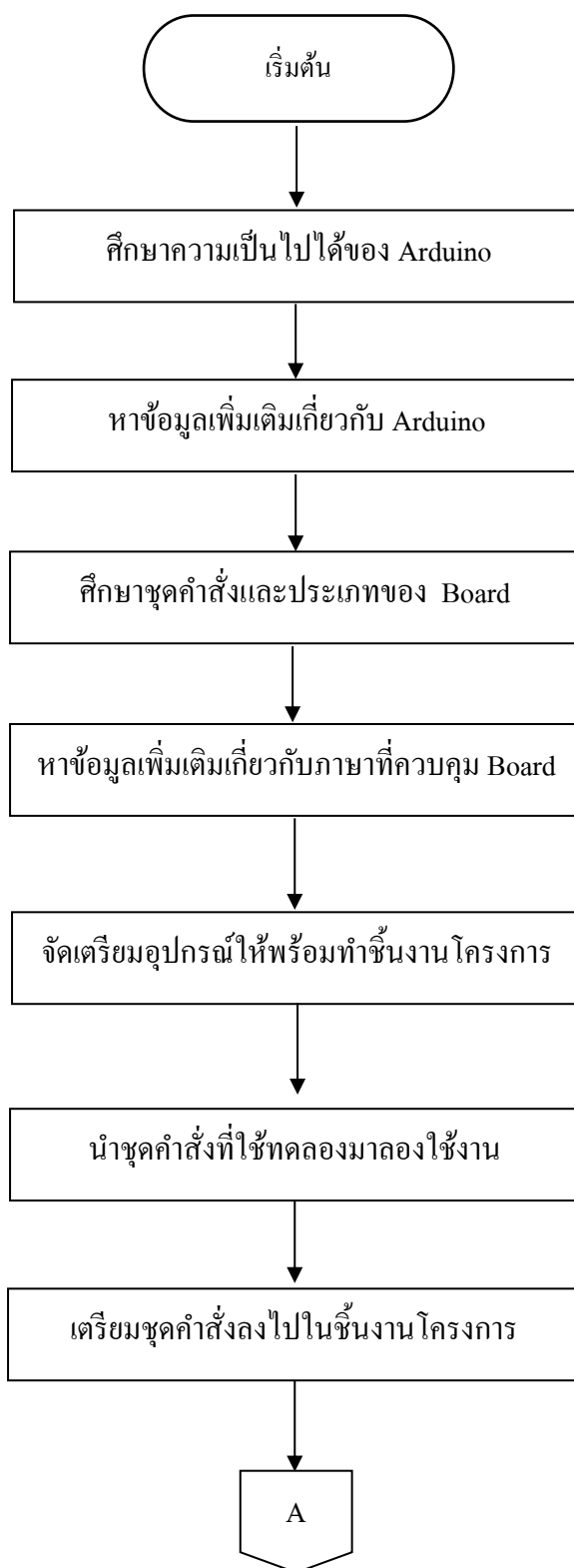
- 13) ทดสอบชิ้นงานโครงการ
- 14) แก้ไขจุดบกพร่องในส่วนต่างที่เกิดข้อผิดพลาด
- 15) ส่งให้อาจารย์ที่ปรึกษาตรวจ

ตารางที่ 3.1 แสดงแผนการดำเนินโครงการ

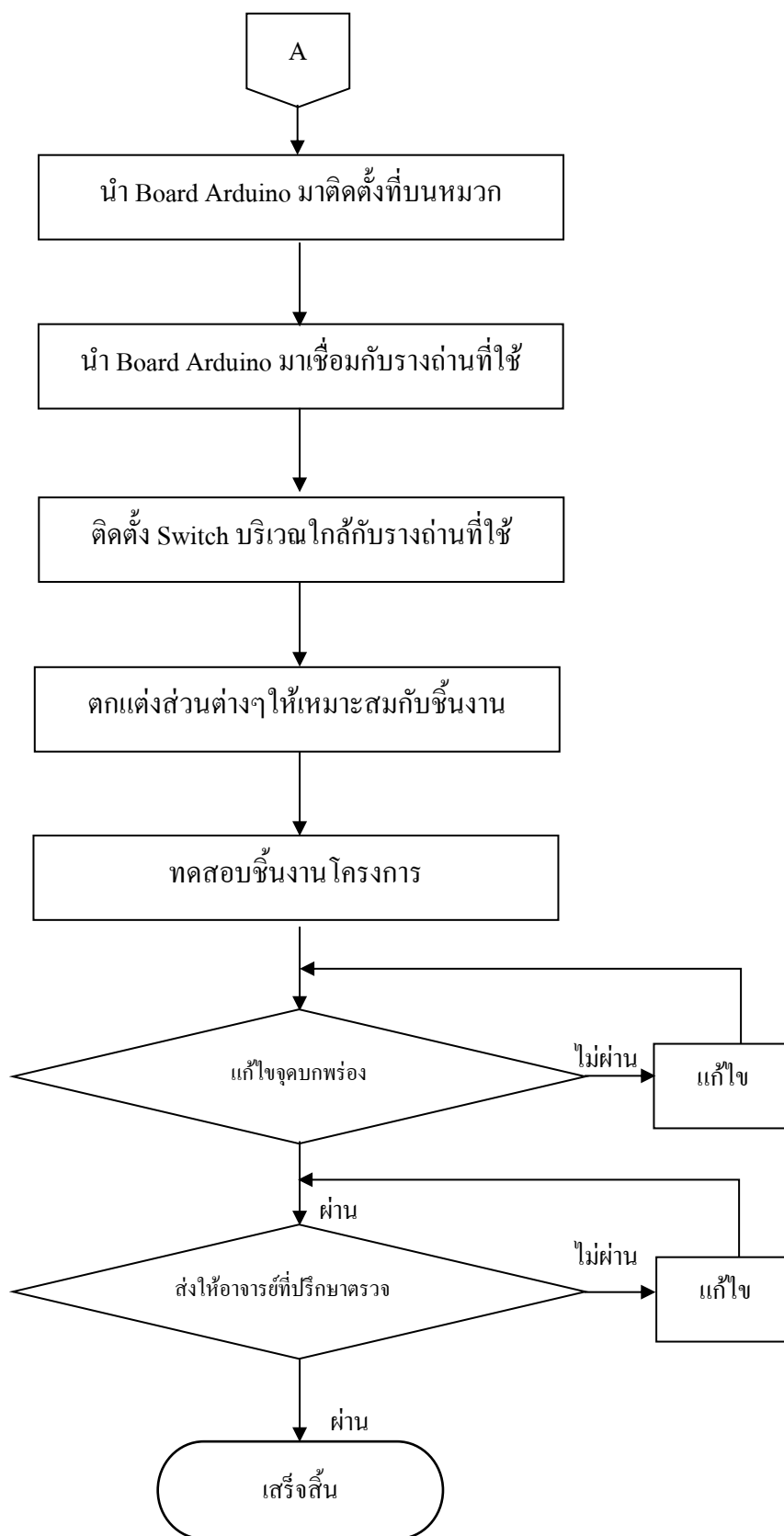
| ลำดับขั้นตอนการทำงาน   | เดือนที่    |            |            |            |            |            |            |            |            |
|--|-------------|------------|------------|------------|------------|------------|------------|------------|------------|
|  | มิ.ย.<br>62 | ก.ค.<br>62 | ส.ค.<br>62 | ก.ย.<br>62 | ต.ค.<br>62 | พ.ย.<br>62 | ธ.ค.<br>62 | ม.ค.<br>63 | ก.พ.<br>63 |
| 1) ศึกษาความเป็นไปได้ของชุดคำสั่ง Arduino                      | *           |            |            |            |            |            |            |            |            |
| 2) หาข้อมูลเพิ่มเติมเกี่ยวกับ Arduino                          | *           |            |            |            |            |            |            |            |            |
| 3) ศึกษาชุดคำสั่งและประเภทของ Board Arduino แต่ละรุ่น          |             | *          |            |            |            |            |            |            |            |
| 4) หาข้อมูลเพิ่มเติมเกี่ยวกับการใช้ควบคุม Board                |             | *          |            |            |            |            |            |            |            |
| 5) จัดเตรียมอุปกรณ์ให้พร้อมทำชิ้นงานโครงการ                    |             |            | *          |            |            |            |            |            |            |
| 6) นำชุดคำสั่งที่ใช้ทดลองมาลองใช้งานและแก้ไขก่อนจัดชิ้นงานจริง |             |            | *          |            |            |            |            |            |            |
| 7) เตรียมชุดคำสั่งลงไปชิ้นงานโครงการ                           |             |            | *          | *          |            |            |            |            |            |
| 8) นำชุดคำสั่งให้ลงไป Board Arduino                            |             |            |            | *          |            |            |            |            |            |
| 9) นำ Board Arduino มาติดตั้งที่บนหมวก                         |             |            | *          | *          |            |            |            |            |            |
| 10) นำ Board Arduino มาเชื่อมกับรางถ่านที่ใช้กระแสไฟ 9.0 V     |             |            | *          |            |            |            |            |            |            |
| 11) ติดตั้ง Switch บริเวณใกล้กับรางถ่านที่ใช้กระแสไฟ 9.0 V     |             |            | *          |            |            |            |            |            |            |

ตารางที่ 3.1 แสดงแผนการดำเนินโครงการ (ต่อ)

| ลำดับขั้นตอนการทำงาน                           | เดือนที่ |      |      |      |      |      |      |      |      |
|--|----------|------|------|------|------|------|------|------|------|
|  | มิ.ย.    | ก.ค. | ส.ค. | ก.ย. | ต.ค. | พ.ย. | ธ.ค. | ม.ค. | ก.พ. |
|  | 62       | 62   | 62   | 62   | 62   | 62   | 62   | 63   | 63   |
| 12) ตกแต่งส่วนต่างให้เหมาะสมกับชิ้นงาน         |          |      |      | *    |      |      |      |      |      |
| 13) ทดสอบชิ้นงานโครงการ                        |          |      |      | *    | *    |      |      |      |      |
| 14) แก้ไขจุดบกพร่องในส่วนต่างที่เกิดข้อผิดพลาด |          |      |      | *    |      |      |      |      |      |
| 15) ส่งให้อาจารย์ที่ปรึกษาตรวจ                 |          |      |      |      | *    |      |      |      |      |



รูปที่ 3.2 แผนภาพ Flowchart ขั้นตอนการดำเนินงาน



รูปที่ 3.2 แผนภาพ Flowchart ขั้นตอนการดำเนินงาน

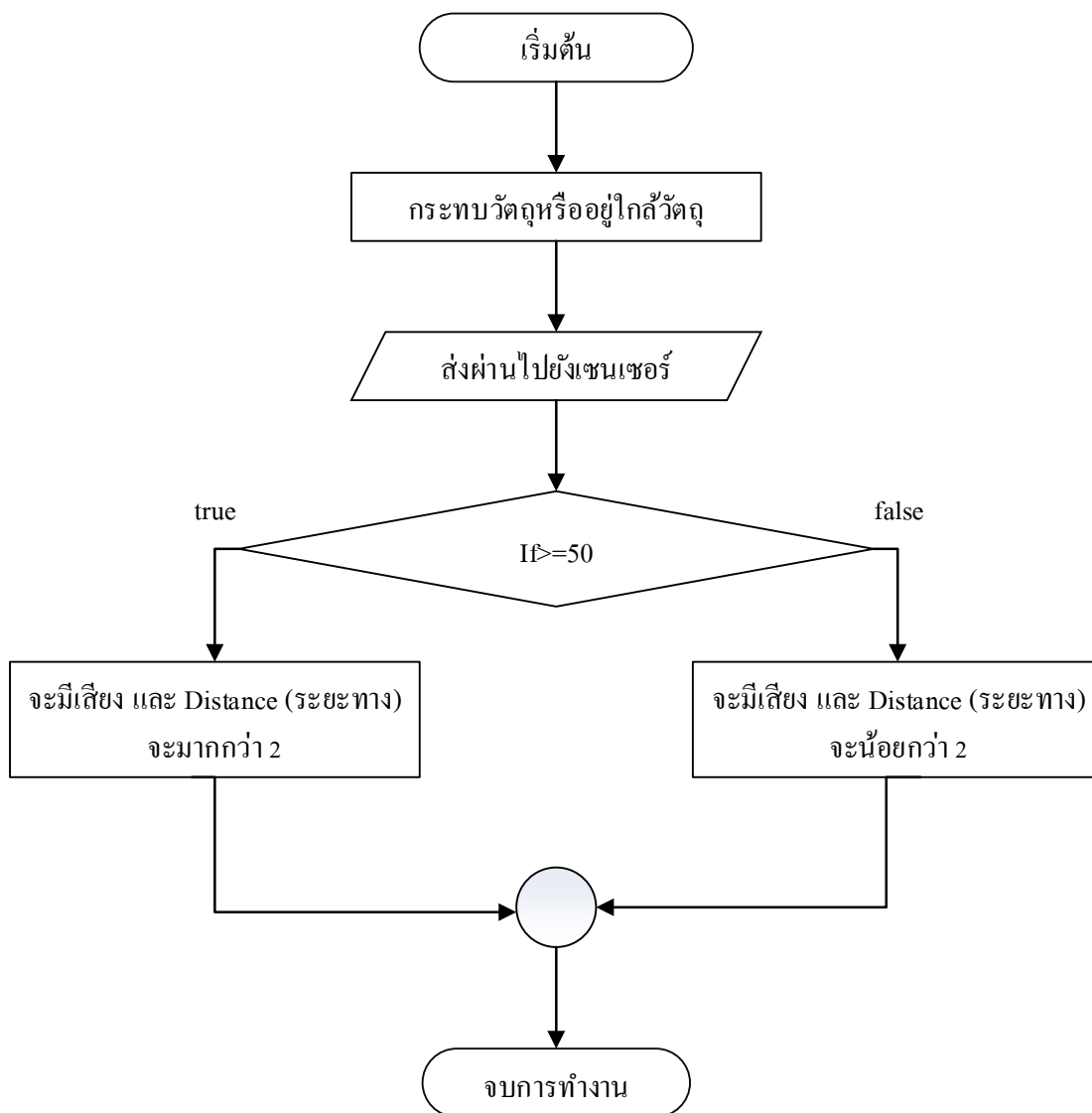


### 3.2 ระบบการทำงานของชิ้นงาน

ในการเขียนชุดคำสั่งชิ้นงานโครงการ จะต้องคิดและวิเคราะห์รูปแบบการทำงานและความเป็นไปได้ของชิ้นงานก่อน ซึ่งมีหลักการทำงานดังนี้

#### 3.3.1 รูปแบบการทำงานเซนเซอร์ตัวที่ 1

มีลำดับขั้นตอนในการทำงาน Arduino จะสามารถทำงานได้ใช้กระแสไฟขนาด 9.0 V และกระแสไฟฟ้าจะทำงานควบคู่กับ Board Arduino และเซนเซอร์ Ultrasonic ตัวที่ 1 จะอยู่บนหมวกและจะรับคำสั่งจากชุดคำสั่งเมื่อเจอสิ่งกีดขวางด้านหน้า



รูปที่ 3.3 แสดง Flowchart รูปแบบการทำงานของวงจร

### 3.3 การสร้างชุดคำสั่ง

การทำแบบจำลองตัวชิ้นงาน กลุ่มของคณะผู้จัดทำได้ใช้โปรแกรม Arduino C++ และใช้ภาษา C++ ในการเขียนชุดคำสั่งลงไปบนชิ้นงานโครงการ

#### 3.3.1 ขั้นตอนการเขียนชุดคำสั่ง

```
/viral science

//Ultrasonic Distance alarm

const int trigPin = 9;

const int echoPin = 10;

const int buzzer = 11;

const int ledPin = 13;


// defines variables

long duration;

int distance;

int safetyDistance;


void setup() {

pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output

pinMode(echoPin, INPUT); // Sets the echoPin as an Input

pinMode(buzzer, OUTPUT);

pinMode(ledPin, OUTPUT);

Serial.begin(9600); // Starts the serial communication

}


void loop() {

// Clears the trigPin
```

```

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

// Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);

// Calculating the distance
distance= duration*0.034/2;

safetyDistance = distance;
if (safetyDistance <= 50){ //Enter the Distance
    digitalWrite(buzzer, HIGH);
    digitalWrite(ledPin, HIGH);
}
else{
    digitalWrite(buzzer, LOW);
    digitalWrite(ledPin, LOW);
}

// Prints the distance on the Serial Monitor
Serial.print("Distance: ");
Serial.println(distance);
}

3

```

## บทที่ 4

### ผลการศึกษา

การจัดทำโครงการหมวดเซนเซอร์ตรวจสอบสิ่งกีดขวางสำหรับผู้พิการทางสายตา มีวัตถุประสงค์เพื่อช่วยลดปัญหาการเกิดอุบัติเหตุแก่ผู้พิการทางสายตาทำให้ผู้พิการทางสายตาได้เกิดความปลอดภัยในชีวิตประจำวัน ซึ่งผลรับที่ได้จากการศึกษาทำให้ได้การดำเนินงานโครงการ

4.1 ขั้นตอนการออกแบบชิ้นงาน

4.2 ขั้นตอนการทำงาน

4.3 ขั้นตอนการทดสอบ

4.4 ผลการทดลอง

#### 4.1 ขั้นตอนการออกแบบชิ้นงาน

ทางคณะผู้จัดทำได้วางแผนการทำงานภายในกลุ่มและแสดงความคิดเห็นกันเพื่อแบ่งหน้าที่การทำงานเพื่อให้เกิดการเป็นแบบแผนและระบบภายในกลุ่ม

4.1.1 แนวคิด เริ่มจากการสังเกตผู้พิการทางสายตาว่าประสบปัญหาอย่างไร ผู้พิการจะใช้ประสาทสัมผัสโดยการใช้ไม้เท้า ทางคณะผู้จัดทำจึงเห็นว่าถ้าเปลี่ยนจากการใช้ไม้เท้าเป็นหมวดที่มีเสียงเตือน ความรู้สึกจะไวกว่าการใช้การไม้เท้า

4.1.2 การเริ่มต้นคิดแบบ การเริ่มต้นคิดทางคณะผู้จัดทำเล็งเห็นว่าผู้พิการทางสายตามีปัญหาต่อการมองเห็นจึงคิดหมวดเซนเซอร์ตรวจสอบสิ่งกีดขวางสำหรับผู้พิการทางสายตาที่ทางคณะผู้จัดทำได้สังเกตและตั้งใจไว้

#### 4.2 ขั้นตอนการทำงาน

ขั้นตอนการทำงานนั้นกล่าวถึงขั้นตอนในการทำงานตั้งแต่ต้นว่ามีขั้นตอนอะไรบ้าง

4.2.1 ศึกษาความเป็นไปได้ของชุดคำสั่ง Arduino

4.2.2 หาข้อมูลเพิ่มเติมเกี่ยวกับชุดคำสั่ง

4.2.3 ศึกษาชุดคำสั่งและประเภทของ Board Arduino แต่ละรุ่น

4.2.4 หาข้อมูลเพิ่มเติมเกี่ยวกับการใช้ภาษา C++ ควบคุมคำสั่ง

4.2.5 จัดเตรียมชุดคำสั่งให้พร้อมทำชิ้นงานโครงการ

4.2.6 นำชุดคำสั่งที่ใช้ทดลองมาลงใช้งานและแก้ไขก่อนจัดชิ้นงานจริง

4.2.7 เริ่มเขียนชุดคำสั่งลงไปในชิ้นงานโครงการ

4.2.8 เขียนชุดคำสั่งให้ลงไปใน Board Arduino

- 4.2.9 นำ Board Arduino มาติดตั้งที่บนหมวก
- 4.2.10 นำ Board Arduino มาเชื่อมกับรางถ่านที่ใช้กระแสไฟ 9.0 V
- 4.2.11 ติดตั้ง Switch บริเวณใกล้กับรางถ่านที่ใช้กระแสไฟ 9.0 V
- 4.2.12 ตกแต่งส่วนต่างให้เหมาะสมกับชิ้นงาน
- 4.2.13 ทดสอบชิ้นงานโครงการ
- 4.2.14 แก้ไขจุดบกพร่องในส่วนต่างที่เกิดข้อผิดพลาด
- 4.2.15 ส่งให้อาจารย์ที่ปรึกษาตรวจ

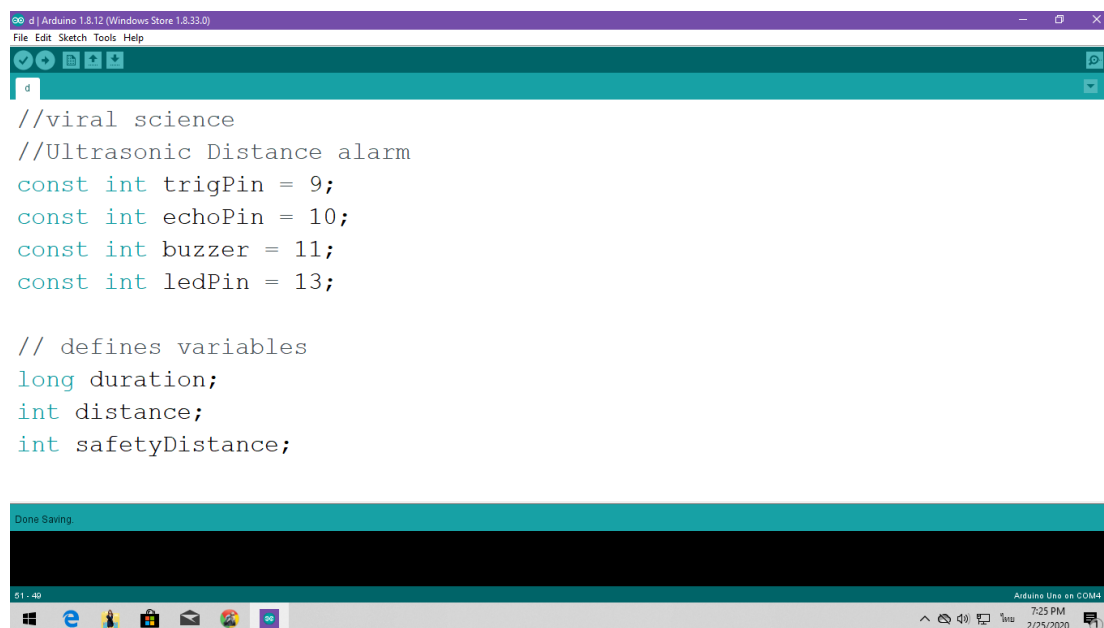
### 4.3 ขั้นตอนการทดสอบ

คณะผู้จัดทำได้ทำการใช้ Switch เพื่อเปิดและการทดสอบโดยการนำเซนเซอร์ไปโดนวัตถุหลายครั้งทำให้เกิดเสียงเตือนในการทดสอบชิ้นงาน

- 4.3.1 ทดสอบส่วนประกอบต่างๆ โดยการกด Switch ปิด/เปิด หลายๆ ครั้ง
- 4.3.2 ทดสอบความเสถียรในการทำงานของอุปกรณ์โดยนำเซนเซอร์ Ultrasonic ไปโดนวัตถุ หรือ สิ่งของต่างๆ

### 4.4 ผลการทดลองในการใช้โค้ดคำสั่ง

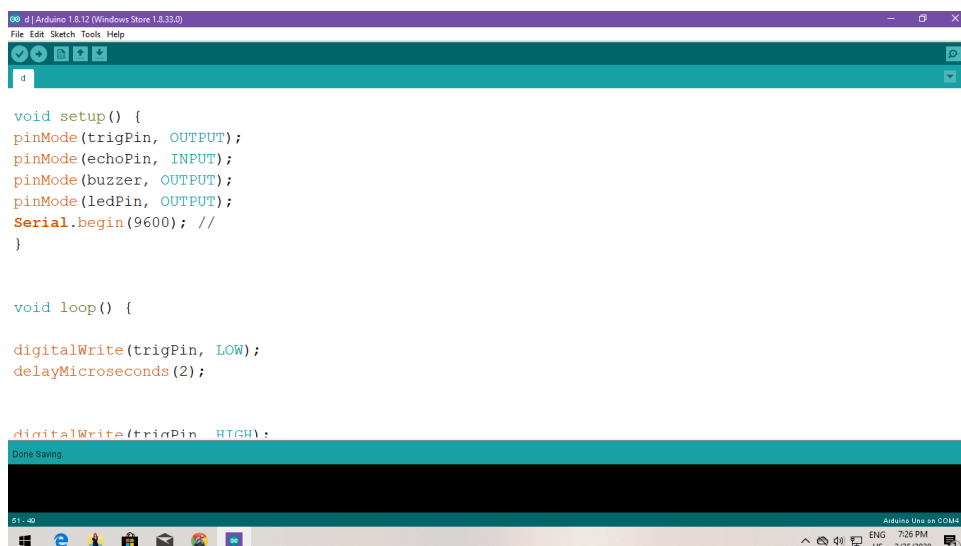
- 4.4.1 การทำงานของ Board Arduino โดยใช้โค้ดคำสั่ง



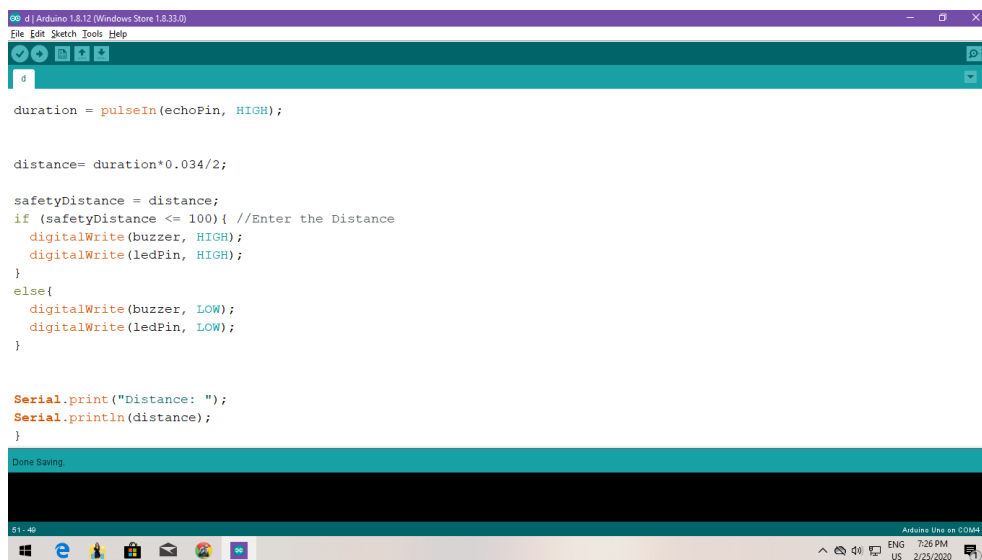
```
//viral science
//Ultrasonic Distance alarm
const int trigPin = 9;
const int echoPin = 10;
const int buzzer = 11;
const int ledPin = 13;

// defines variables
long duration;
int distance;
int safetyDistance;
```

รูปที่ 4.1 การพิมพ์คำสั่งโค้ด Arduino ประกาศค่าของ Board Arduino



รูปที่ 4.2 การพิมพ์คำสั่งโค้ด Arduino รับค่าจาก เซนเซอร์ Ultrasonic



รูปที่ 4.3 การพิมพ์คำสั่งโค้ด Arduino คำนวณระยะทางจาก Ultrasonic

## บทที่ 5

### สรุป อภิปรายผล และข้อเสนอแนะ

ในการสร้างหมวกเซนเซอร์ตรวจจับสิ่งกีดขวางสำหรับผู้พิการทางสายตา นั้นบรรลุดูจุดประสงค์ที่วางไว้ สามารถใช้งานได้อย่างที่ต้องการ แต่การดำเนินงานนั้นประสบปัญหาต่างๆ หลายอย่าง ซึ่งผู้ดำเนินโครงการมีข้อเสนอแนะที่จะนำมาใช้พัฒนาปรับปรุงแก้ไขให้สามารถเตือนผู้พิการทางสายตาสามารถหลบสิ่งกีดขวางเพื่อไม่ให้เกิดอุบัติเหตุ เพื่อมีประสิทธิภาพในการใช้งานมากยิ่งขึ้น

#### 5.1 วัตถุประสงค์ของโครงการ

- 1) เพื่อให้ผู้พิการทางสายตาลดปัญหาและอุปสรรคสำหรับการเดินทาง
- 2) เพื่อให้ผู้จัดทำโครงการนำความรู้จากการศึกษาในหลักสูตรของประกาศนียบัตรวิชาชีพ มาประยุกต์ใช้ในการทำโครงการ
- 3) เพื่อลดการอุบัติเหตุแก่ผู้พิการทางสายตา

#### 5.2 ประโยชน์ที่ได้รับจากการศึกษา

- 1) เพื่อให้ผู้พิการทางสายตาลดปัญหาและอุปสรรคสำหรับการเดินทาง
- 2) เพื่อให้ผู้จัดทำโครงการนำความรู้จากการศึกษาในหลักสูตรของประกาศนียบัตรวิชาชีพ มาประยุกต์ใช้ในโครงการ
- 3) เพื่อลดการอุบัติเหตุแก่ผู้พิการทางสายตา

#### 5.3 ปัญหาที่ประสบในการดำเนินโครงการ

มีการบกพร่องในการสั่งซื้อสินค้าในการทำโครงการ เพราะเนื่องจากบางอุปกรณ์เลือกแล้วแต่ไม่ได้ทำการกดสั่งซื้อจึงทำให้ไม่ได้รับอุปกรณ์ตามที่กำหนดระยะเวลาการส่งช้าสุดประมาณ 3-7 วันจึงทำให้ผู้จัดทำเกิดความล่าช้าต่อการทำชิ้นงานโครงการ เครื่องมือเกิดการชำรุดหรือเสียหายจึงทำให้ไม่สามารถใช้งานร่วมกันได้ ประสบการณ์น้อย เพราะการทำงานจำเป็นต้องอาศัยประสบการณ์เป็นหลัก คณะผู้จัดทำผ่านการทำงานเป็นกลุ่มและประดิษฐ์ชิ้นงานไม่ได้มากต้องศึกษาและทดลอง

## 5.4 ผลการดำเนินโครงการ

การดำเนินโครงการหมวดเซนเซอร์ตรวจจับสิ่งกีดขวางสำหรับผู้พิการทางสายตา เริ่มตั้งแต่การนำเสนอต่อคณะกรรมการพิจารณา ทางคณะกรรมการได้ให้หาข้อมูลเพิ่มเติมเรียบร้อย จึงได้รับการอนุมัติจากคณะกรรมการในการจัดสร้าง จึงมาศึกษาข้อมูลจากทางอินเทอร์เน็ต จึงมาวางแผนการดำเนินโครงการ โดยทำการหาอุปกรณ์ที่สามารถนำมาใช้ร่วมกันได้

ผลการดำเนินโครงการหมวดตรวจจับสิ่งกีดขวางสำหรับผู้พิการทางสายตาสามารถตรวจจับสิ่งกีดขวางและมีเสียงเตือนและสั่นได้

## 5.5 อภิปรายผล

จากผลการดำเนินโครงการนี้ถือว่าประสบความสำเร็จตามที่ตั้งจุดประสงค์ไว้คือ ชุดคำสั่งสามารถใช้กับหมวดเซนเซอร์ตรวจจับสิ่งกีดขวางสำหรับผู้พิการทางสายตาได้ดี และมีประสิทธิภาพในการใช้งาน หมวดเซนเซอร์ตรวจจับสิ่งกีดขวางสำหรับผู้พิการทางสายตาส่งขึ้นและนำไปประโยชน์ได้เป็นอย่างดีสำหรับผู้พิการทางสายตา นอกจากนี้ คณะผู้จัดทำยังได้รับความรู้และประสบการณ์การทำโครงการนี้เป็นอย่างมากซึ่งสอดคล้องกับผลงานวิจัยของ T.E Lawrence มีบางหลักฐานระบุว่าเมื่อ 900 ปีก่อนคริสตกาล ทหารเริ่มสวมหมวกกันน็อกที่ทำจากหนังหรือบรอนซ์ เพื่อป้องกันภัยขณะต่อสู้กับข้าศึกและมีการพัฒนาเรื่อยมาจนกระทั่ง เมื่อวันที่ 13 พฤษภาคม ค.ศ.1935 นายT.E Lawrence ได้เสียชีวิตจากอุบัติเหตุจักรยานยนต์ คุณหมอ Huge Calms เจ้าของคนไข้ที่คอยดูแลอาการของเขารู้สึกเสียใจเป็นอย่างมากที่เขาได้เสียชีวิต จึงนำไปสู่การวิจัยและพัฒนาให้มีหมวกกันน็อก เพื่อป้องกันอันตรายอย่างจริงจัง ในปี ค.ศ. 1935 จนถึงปัจจุบันผู้คนต่างให้ความสำคัญกับการใช้หมวกกันน็อกช่วยลดอุบัติเหตุ

## 5.6 ข้อเสนอแนะ

### 5.6.1 ข้อเสนอแนะทั่วไป

- 1) สามารถใช้งานในสถานที่ที่มีคนจำนวนมากได้
- 2) สามารถเชื่อมต่อกับอุปกรณ์ชนิดอื่น
- 3) ควรมีลูกเล่นในการออกแบบให้น่าใช้งาน

### 5.6.2 ข้อเสนอแนะทางเทคนิค

- 1) ควรทำชุดคำสั่งให้เข้าใจมากยิ่งขึ้น
- 2) ควรเพิ่มซอฟต์แวร์เข้ามาช่วยทำให้ดูน่าใช้มากขึ้น



## บรรณานุกรม

- ธนวัฒน์ ตลับทอง. (2560). การใช้ if else. ค้นข้อมูลวันที่ 3 กันยายน 2562, จาก <http://marcuscode.com/lang/c/flow-control>
- ร้านค้าออนไลน์ Arduinoall . (2562). วิธีการติดตั้ง Driver Arduino. ค้นข้อมูลวันที่ 3 กันยายน 2562, จาก <http://arduinoall.com/article/201/1-สอน-arduino-วิธีติดตั้งโปรแกรม-arduino-ide>
- วิกิพีเดีย สารานุกรมเสรี. (2562). ความต้องการของระบบ. ค้นข้อมูลวันที่ 3 กันยายน 2562, จาก <https://th.wikipedia.org/wiki/>
- วิกิพีเดีย สารานุกรมเสรี. (2562). ภาษาซี. ค้นข้อมูลวันที่ 3 กันยายน 2562, จาก <https://th.wikipedia.org/wiki/>
- วิกิพีเดีย สารานุกรมเสรี. (2562). สอนการใช้ Arduino. ค้นข้อมูลวันที่ 3 กันยายน 2562, จาก <https://th.wikipedia.org/wiki/>
- อินทนนท์ ปัญญาโสภา. (2562). การคำนวณระยะทาง. ค้นข้อมูลวันที่ 3 กันยายน 2562, จาก <https://www.grappik.com/10-step-how-to-create-logo>

## ภาคผนวก

ภาคผนวก ก แบบเสนอร่างโครงการ

ภาคผนวก ข ความคืบหน้า

ภาคผนวก ค คู่มือการให้งาน

ภาคผนวก ง ประวัติผู้จัดทำโครงการ

ภาคผนวก ก  
แบบเสนอโครงร่าง

ภาคผนวก ข  
ความคืบหน้า

ภาคผนวก ค  
คู่มือการใช้งาน

ภาคผนวก ง  
ประวัติผู้จัดทำ



### ประวัติผู้วิจัย

ชื่อ-ชื่อสกุล

นายจิราเมธ ประเทือง

วันเดือนปีเกิด

27 สิงหาคม 2544

สถานที่เกิด

สมุทรปราการ

สถานที่อยู่ปัจจุบัน

50/2 หมู่ที่ 5 ถ.ปู่เจ้าสมิงพราย ต.สำโรงกลาง

อ.พระประแดง จ.สมุทรปราการ

ประวัติการศึกษา

พ.ศ. 2557

มัธยมศึกษาปีที่ 3 โรงเรียนสรรพาวุธวิทยา

พ.ศ. 2560

ประกาศนียบัตรวิชาชีพ

สาขาวิชาเทคโนโลยีสารสนเทศ

วิทยาลัยเทคโนโลยีอรรณพวิทยพัฒน์

ผลงานและกิจกรรม

พ.ศ. 2560-2562

เข้าร่วมการแข่งขันทักษะวิชาชีพ สมาคมวิทยาลัย  
เทคโนโลยีและอาชีวศึกษาเอกชนแห่งประเทศไทย  
ในพระราชูปถัมภ์สมเด็จพระเทพรัตนราชสุดา  
สยามบรมราชกุมารี ประจำปีการศึกษา 2560-2562



### ประวัติผู้วิจัย

ชื่อ-ชื่อสกุล

นายพลศวัจน์ คำคง

วันเดือนปีเกิด

16 กุมภาพันธ์ 2545

สถานที่เกิด

สมุทรปราการ

สถานที่อยู่ปัจจุบัน

369/216 ต.เทพารักษ์ อ.เมือง จ.สมุทรปราการ

ประวัติการศึกษา

พ.ศ.2557

โรงเรียนสมุทรปราการ

พ.ศ.2560

ประกาศนียบัตรวิชาชีพ

สาขาวิชาเทคโนโลยีสารสนเทศ

วิทยาลัยเทคโนโลยีอรรณพวิทยพัฒน์วิชาการ





### ประวัติผู้วิจัย

ชื่อ-ชื่อสกุล

นายพชรดนัย โอบฐ์ประไพ

วันเดือนปีเกิด

30 มิถุนายน 2545

สถานที่เกิด

สมุทรปราการ

สถานที่อยู่ปัจจุบัน

1172 หมู่1 ซอย แบริง 48 ถนน สุขุมวิท

ตำบลตำโรงเหนือ อำเภอเมือง จังหวัดสมุทรปราการ

ประวัติการศึกษา

พ.ศ.2557

โรงเรียนลาซาลกรุงเทพ

พ.ศ.2560

ประกาศนียบัตรวิชาชีพ

สาขาวิชาเทคโนโลยีสารสนเทศ

วิทยาลัยเทคโนโลยีอรรถวิทย์พัฒนวิชาการ