



ชุดคำสั่งควบคุมมอเตอร์โต๊ะเก็บโน้ตบุ๊กอิเล็กทรอนิกส์

Control Motor Code ET-Table

จัดทำโดย

นายภาณุพล สอนน่วม

นางสาวยุวรินทร์ เศรษฐนิโรจน์

นางสาวรัตนภรณ์ ทองล้ำ

โครงการนี้เป็นส่วนหนึ่งของวิชาโครงการ หลักสูตรประกาศนียบัตรวิชาชีพ

สาขาวิชาเทคโนโลยีสารสนเทศ

วิทยาลัยเทคโนโลยีอรรถวิทย์พัฒนการ

ปีการศึกษา 2562

ชุดคำสั่งควบคุมมอเตอร์โตะเก็บน้ำตักอิเล็กทรอนิกส์

Control Motor Code ET-Table

จัดทำโดย

นายภาณุพล	สอนน่วม
นางสาวยุวรินทร์	เศรษฐนิโรจน์
นางสาวรัตนภรณ์	ทองคำ

โครงการนี้เป็นส่วนหนึ่งของวิชาโครงการ หลักสูตรประกาศนียบัตรวิชาชีพ

สาขาวิชาเทคโนโลยีสารสนเทศ

วิทยาลัยเทคโนโลยีอรรณพวิทยพัฒน์

ปีการศึกษา 2562

COPYRIGHT 2019

COLLEGE OF INFORMATION TECHNOLOGY

ATTAWIT COMMERCIAL TECHNOLOGY COLLEGE



ชื่อโครงการภาษาไทย

ชุดคำสั่งควบคุมมอเตอร์ไต้ะเก็บไน้ตบู้คืเล็กทรอนิกส์

ชื่อโครงการภาษาอังกฤษ

Control Motor Code ET-Table

โดย	1. นายภาณุพล	สอนน่วม	รหัสประจำตัว 39464
	2. นางสาวยุวรินทร์	เศรษฐนิโรจน์	รหัสประจำตัว 39797
	3. นางสาวรัตนกรณั	ทองถ้ำ	รหัสประจำตัว 40147

.....

คณะกรรมการอนุมัติให้เอกสารโครงการฉบับนี้เป็นส่วนหนึ่งของการศึกษา วิชาโครงการตามหลักสูตรประกาศนียบัตร (ปวช.) สาขาวิชาเทคโนโลยีสารสนเทศ วิทยาลัยเทคโนโลยีอรรถวิทย์พัฒนการ (ATC)

.....

(อาจารย์ คุณานนท์ สุขเกษม)

อาจารย์ที่ปรึกษา

.....

(อาจารย์คุณานนท์ สุขเกษม)

หัวหน้าสาขาวิชาเทคโนโลยีสารสนเทศ

บทคัดย่อ

หัวข้อโครงการ	ชุดคำสั่งควบคุมมอเตอร์ไต่เก็บโน้ตบุ๊กอิเล็กทรอนิกส์ Control Motor Code ET-Table		
ผู้จัดทำโครงการ	1. นายภาณุพล	สอนน่วม	รหัสประจำตัว 39464
	2. นางสาวยุวรินทร์	เศรษฐนิโรจน์	รหัสประจำตัว 39797
	3. นางสาวรัตนากรณ์	ทองคำ	รหัสประจำตัว 40147
อาจารย์ที่ปรึกษา	อาจารย์คุณานนท์ สุขเกษม		
สาขาวิชา	เทคโนโลยีสารสนเทศ		
สถาบัน	วิทยาลัยเทคโนโลยีอรรถวิทย์พัฒนศึกษา ปีการศึกษา 2562		

บทคัดย่อ

นักศึกษาได้จัดทำโครงการขึ้นนี้ โดยได้นำความรู้จากการศึกษาที่เรียนมาตลอดเวลา 3 ปี เพื่อใช้ความรู้ทั้งหมดจัดทำโครงการนี้ขึ้น โดยมีการวางแผน หาข้อมูล หาความรู้ เพื่อประกอบกับการจัดทำโครงการนี้ขึ้น

คณะผู้จัดทำโครงการได้เห็นถึงปัญหาเรื่องพื้นที่ในการทำงานและความปลอดภัยของโน้ตบุ๊ก จึงหาวิธีการในการแก้ไขปัญหา และได้จัดทำโครงการขึ้นมีชื่อว่า ไต่เก็บโน้ตบุ๊กอิเล็กทรอนิกส์ การทำงานของชิ้นงานก็จะทำการเก็บโน้ตบุ๊กเมื่อเราไม่ได้ใช้งาน และรักษาโน้ตบุ๊กไว้เพื่อความปลอดภัย

โครงการประสบความสำเร็จไปได้ด้วยดี สามารถเก็บโน้ตบุ๊กได้ตามที่คิดไว้ การทำงานเป็นไปอย่างปกติ และมีแนวทางในการพัฒนาต่อ โดยเพิ่มความปลอดภัยกับตัวโน้ตบุ๊ก ให้เปลี่ยนจากปุ่มกดสัมผัส เป็น ปุ่มกดแบบสวิตช์ผ่านเพื่อ รักษาโน้ตบุ๊กและตัวซอฟต์แวร์ข้อมูลภายในเครื่อง และเปลี่ยนวัสดุที่ใช้เพื่อให้ไต่มีน้ำหนักที่เบาขึ้น

กิตติกรรมประกาศ

โครงการของชุดคำสั่งควบคุมมอเตอร์ไต่เก็บโน้ตบุ๊กอิเล็กทรอนิกส์ของนักศึกษาวิทยาลัยเทคโนโลยีอรรถวิทย์พัฒนการสำเร็จลุล่วงไปได้ด้วยดี

งานวิจัยฉบับนี้สำเร็จได้ด้วยความกรุณาของคณะอาจารย์ทุกท่านในสาขาวิชาเทคโนโลยีสารสนเทศ ที่เสียสละเวลา ให้คำแนะนำในการค้นคว้าข้อมูลและความรู้ในด้านต่างๆ ที่สามารถนำมาประยุกต์ใช้กับงานได้อย่างดี และขอขอบคุณวิทยาลัยเทคโนโลยีอรรถวิทย์พัฒนการที่ทำให้มีโครงการดีๆ ที่ทำให้มีความคิดริเริ่มสร้างสิ่งประดิษฐ์นวัตกรรมนี้ขึ้นมา

พระคุณเหนือสิ่งใดที่ต้องกล่าวขอบพระคุณไว้คือ คุณพ่อคุณแม่และครอบครัวญาติพี่น้อง และคนที่รักเป็นอย่างยิ่งที่คอยให้การสนับสนุนเลี้ยงดู ส่งเสีย มอบความรักความอบอุ่นให้เป็นอย่างดีและอยู่ข้างกันตลอดมา

สุดท้ายขอขอบคุณนักศึกษาวิทยาลัยเทคโนโลยีอรรถวิทย์พัฒนการที่กรุณาให้ความร่วมมือในการตอบแบบสอบถามได้เป็นอย่างดี ตลอดจนขอขอบคุณเพื่อนๆ ทุกคนที่มีส่วนร่วม ลงมือลงแรงทำงานประสบความสำเร็จ ทำให้ผ่านอุปสรรคต่างๆ ในการจัดทำโครงการนี้ไปได้ด้วยดี กระทั่งบรรลุผลสำเร็จในที่สุด คณะผู้วิจัยขอขอบคุณมา ณ โอกาสนี้

นายภาณุพล สอนน่วม

นางสาวยุรินทร์ เศรษฐนิโรจน์

นางสาวรัตนภรณ์ ทองล้ำ

สารบัญ

หน้า

บทคัดย่อภาษาไทย	I
กิตติกรรมประกาศ.....	II
สารบัญ	III
สารบัญตาราง	IX
สารบัญภาพ	X
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของโครงการ.....	2
1.2.1 เพื่อสร้างโต๊ะสำหรับเก็บโน้ตบุ๊กใช้งานในพื้นที่จำกัด.....	2
1.2.2 เพื่อจัดเก็บ Notebook ได้อย่างปลอดภัย	2
1.2.3 เพื่อนำความรู้ที่ได้จากการเรียนการสอนในสาขาวิชาเทคโนโลยี สารสนเทศมาประยุกต์ใช้ในการทำโครงการ	2
1.3 ขอบเขตของโครงการ	2
1.3.1 ใช้ Arduino UNO R3 ในการควบคุมระบบ	2
1.3.2 ใช้ตัวขับเคลื่อนมอเตอร์ L298Nในการควบคุมมอเตอร์	2
1.3.3 ใช้สวิตช์ปุ่มกดในการเปิด - ปิด.....	2
1.3.4 ใช้ Switch Power Supply แปลงไฟฟ้าบ้าน จ่ายให้อุปกรณ์	2
1.3.5 รองรับ Notebook ขนาด 15.6 นิ้ว.....	2
1.3.6 โต๊ะสามารถพับขาได้	2
1.3.7 ใช้ไม้อัดในการทำโต๊ะ	2
1.3.8 โต๊ะขนาด	2
1.4 ขั้นตอนการดำเนินงาน	2
1.4.1 คิดหัวข้อโครงการ	2
1.4.2 นำเสนอโครงการ	2
1.4.3 จัดทำแบบเสนอร่างโครงการ	2
1.4.4 ส่งแบบเสนอโครงการ	2
1.4.5 ศึกษาอุปกรณ์ในการทำโต๊ะเก็บโน้ตบุ๊กอัตโนมัติ	2
1.4.6 จัดเตรียมอุปกรณ์ในการทำชิ้นงาน	2
1.4.7 จัดทำเอกสารโครงการ บทที่ 1 และ บทที่ 3	2
1.4.8 ส่งเอกสารโครงการ บทที่ 1 และ บทที่ 3	2

สารบัญ(ต่อ)

	หน้า
1.4.9 เริ่มทำชิ้นงานโครงการ	2
1.4.10 จัดทำเอกสารโครงการ บทที่ 2	2
1.4.11 ส่งเอกสารโครงการ บทที่ 2	2
1.4.12 ทดสอบการทำงานของตัวชิ้นงานโครงการ	2
1.4.13 แก้ไขข้อบกพร่องที่เกิดขึ้นในการทดสอบ	2
1.4.14 เตรียมโครงการสอบต่อคณะกรรมการ	2
1.4.15 สอบโครงการต่อคณะกรรมการ.....	2
1.4.16 แก้ไขตามที่คณะกรรมการแจ้ง	2
1.4.17 ส่งชิ้นงานให้คณะกรรมการตรวจสอบ	2
1.4.18 จัดทำเอกสารโครงการ บทที่ 4-5	2
1.4.19 ส่งเอกสารโครงการ บทที่ 4-5	3
1.4.20 ส่งเล่มโครงการพร้อมชิ้นงานที่สมบูรณ์	3
1.6 ประโยชน์ที่คาดว่าจะได้รับ	9
1.6.1 ได้โต๊ะสำหรับเก็บโน้ตบุ๊กใช้งานในพื้นที่จำกัด	9
1.6.2 จัดเก็บ Notebook ได้อย่างปลอดภัย	9
1.6.3 ได้นำความรู้ที่ได้จากการเรียนการสอนในสาขาวิชาเทคโนโลยีสารสนเทศมา ประยุกต์ใช้ในการทำโครงการ	9
1.7 งบประมาณที่ใช้ในการทำโครงการ	9
1.7.1 Arduino UNO R3	9
1.7.2 L298N motor driver.....	9
1.7.3 Switching 12 V 10 A.....	9
1.7.4 Jump Wire (Male to Female).....	9
1.7.5 ปลั๊กตัวผู้.....	9
1.7.6 มอเตอร์เกียร์ DC 12V	9
1.7.7 สายไฟ.....	9
1.7.8 ไม้อัด.....	9
1.7.9 ปุ่มกด.....	9
1.7.10 เฟืองสะพาน.....	9
1.7.11 เฟืองกลม.....	9
1.7.12 เล่มโครงการ	9

สารบัญ(ต่อ)

	หน้า
1.7.13 ซิดี CD	9
1.7.14 อื่น ๆ	9
บทที่ 2 แนวคิด ทฤษฎี เอกสารและงานวิจัยที่เกี่ยวข้อง	10
2.1 สอนใช้งาน Arduino ควบคุมมอเตอร์ DC หมุนซ้าย หมุนขวา	10
2.2 สอนใช้งาน Arduino if else ตรวจสอบเงื่อนไข ถูกผิด	13
2.3 ภาษาซี (C Programming Language).....	17
2.4 ผังงาน (Flowchart)	33
2.4.1 ประโยชน์ของผังงาน	35
2.4.2 ประเภทของผังงาน	35
2.5 โปรแกรม Arduino	39
2.6 การเขียนโปรแกรมคอมพิวเตอร์ (Computer programming).....	39
2.7 หลักการทำงานของบอร์ด Arduino.....	39
2.7.1 Arduino คืออะไร	39
2.7.2 จุดเด่นที่ทำให้บอร์ด Arduino เป็นที่นิยม.....	40
2.7.2.1 ง่ายต่อการพัฒนามีรูปแบบคำสั่งพื้นฐาน.....	40
2.7.2.2 มี Arduino Community กลุ่มคนที่ร่วมกันพัฒนาที่แข็งแกร่ง.....	40
2.7.2.3 Open Hardware ทำให้ผู้ใช้สามารถนำบอร์ดไปต่อยอด	40
2.7.2.4 ราคาไม่แพง.....	40
2.7.3 ขั้นตอนการอัปโหลดโค้ดลงบอร์ด Arduino	41
2.7.3.1 เขียนโปรแกรมบนคอมพิวเตอร์	
ผ่านทางโปรแกรม ArduinoIDE	41
2.7.3.2 หลังจากเขียนโค้ดโปรแกรมเรียบร้อยแล้ว	41
2.7.4 ข้อมูลของส่วนต่างๆ ในบอร์ด Arduino.....	43
2.7.4.1 USBPort: ใช้สำหรับต่อกับ Computer	
เพื่ออัปโหลดโปรแกรมเข้าMCU และทำให้ไฟให้กับบอร์ด	43
2.7.4.2 Reset Button: เป็นปุ่ม Reset ใช้กดเมื่อต้องการให้ MCU	
เริ่มการทำงานใหม่.....	43
2.7.4.3 ICSP Port ของ Atmega16U2 เป็นพอร์ตที่ใช้โปรแกรม	
Visual Com port.....	43

สารบัญ(ต่อ)

หน้า

2.7.4.4 I/O Port: Digital I/O ตั้งแต่ขา D0 ถึง D13 นอกจากนี้ บาง Pin จะทำหน้าที่อื่นๆ เพิ่มเติมด้วย.....	43
2.7.4.5 ICSP Port: Atmega328 เป็นพอร์ตที่ใช้โปรแกรม Bootloader	43
2.7.4.6 MCU: Atmega328 เป็น MCU ที่ใช้บนบอร์ด Arduino	43
2.7.4.7 I/O Port: นอกจากจะเป็น Digital I/O แล้วยังเปลี่ยนเป็นช่องรับสัญญาณอนาล็อกตั้งแต่ขา A0-A5....	43
2.7.4.8 Power Port: ไฟเลี้ยงของบอร์ดเมื่อต้องการทำให้กับวงจร ภายนอกประกอบด้วยขาไฟเลี้ยง +3.3 V, +5V, GND, Vin	43
2.7.4.9 Power Jack: รับไฟจาก Adapter โดยที่แรงดันอยู่ ระหว่าง 7-12 V40.....	43
2.7.4.10 MCU ของ Atmega16U2 เป็น MCU ที่ทำหน้าที่เป็น USB to Serial โดย Atmega328 จะติดต่อกับ Computer ผ่าน Atmega16U2	43
2.7.5 การทำงานของ Code.....	43
2.7.5.1 CODE ควบคุมการทำงานของ Node MCU / ESP8266	43
2.7.5.2 CODE ควบคุมการทำงานของ LCD Display	46
บทที่ 3 วิธีดำเนินการวิจัย.....	51
3.1 การวางแผนการเขียนชุดคำสั่ง	51
3.1.1 การวางแผนทำโครงการ	51
3.2 ออกแบบการทำงานของระบบ.....	55
3.3 ดำเนินการเขียนชุดคำสั่ง	56
บทที่ 4 ผลการวิจัย	59
4.1 ขั้นตอนการวางแผนการเขียนชุดคำสั่ง.....	59
4.1.1 หาข้อมูลเกี่ยวกับคำสั่ง Arduino	59
4.1.2 ศึกษาเกี่ยวกับการทำงานของมอเตอร์	59
4.1.3 วาด FlowChart การทำงานของระบบ	59
4.1.4 เตรียมอุปกรณ์ที่ใช้ทดลอง มอเตอร์ สายไฟ สวิตช์ บอร์ด	59
4.1.5 เริ่มเขียนชุดคำสั่ง	59
4.1.6 ต่ออุปกรณ์ต่างๆ ที่ใช้ทดลอง.....	59

สารบัญ(ต่อ)

	หน้า
4.1.7 ทดสอบระบบการทำงาน	59
4.1.8 นำCodeที่สมบูรณ์ไปใช้ร่วมกับโตะ	59
4.1.9 ทดสอบการทำงาน	59
4.1.10 แก้ไขข้อผิดพลาด	59
4.2 ขั้นตอนการดำเนินการเขียนชุดคำสั่ง	59
4.2.1 ศึกษาการเขียนโปรแกรม	59
4.2.2 เขียนโปรแกรมซอฟต์แวร์ Arduino IDE	59
4.2.3 อัปโหลดโปรแกรมลง บอร์ด Arduino IDE หรือบอร์ดตัวอื่นๆ ที่คล้ายกัน	59
4.3 ขั้นตอนการทดสอบ	60
บทที่ 5 สรุปผลการวิจัยอภิปรายผลและข้อเสนอแนะ	62
5.1 วัตถุประสงค์ของโครงการ	62
5.1.1 เพื่อสร้างโตะสำหรับเก็บโน้ตบุ๊กใช้งานในพื้นที่จำกัด	62
5.1.2 เพื่อจัดเก็บ Notebook ได้อย่างปลอดภัย	62
5.1.3 เพื่อนำความรู้ที่ได้จากการเรียนการสอนในสาขาวิชาเทคโนโลยีสารสนเทศ มาประยุกต์ใช้ในการทำโครงการ	62
5.2 ประโยชน์ที่ได้รับจากการศึกษา	62
5.2.1 ได้โตะญี่ปุ่นใช้งานในพื้นที่จำกัด	62
5.2.2 จัดเก็บ Notebook ได้อย่างปลอดภัย	62
5.2.3 เพื่อนำความรู้ที่ได้จากการเรียนการสอนในสาขาวิชาเทคโนโลยีสารสนเทศ มาประยุกต์ใช้ในการทำโครงการ	62
5.3 ปัญหาที่ประสบในการดำเนินโครงการ	63
5.3.1 ปัญหาด้านการทำงานของโตะเก็บโน้ตบุ๊กอัตโนมัติ เนื่องจากเกิดความ ผิดพลาดจากเรื่องระยะเวลาในการเปิดหรือปิด และแรงของมอเตอร์ น้ำหนักของโน้ตบุ๊กทำให้เกิดการคลาดเคลื่อน	63
5.3.2 ปัญหาด้านวงจรสายไฟเชื่อมต่อกันไม่แน่นหรือหลุดทำให้ มอเตอร์ไม่ทำงาน จึงทำให้ระบบเกิดการขัดข้อง	63
5.3.3 ปัญหาด้านวงจรสายไฟเชื่อมต่อกันไม่แน่นหรือหลุดทำให้ มอเตอร์ไม่ทำงาน	63
5.4 ผลการดำเนินโครงการ	63
5.5 อภิปรายผล	63

สารบัญ(ต่อ)

	หน้า
5.6 ข้อเสนอแนะ	63
5.6.1 ข้อเสนอทั่วไป.....	63
5.6.1.1 แนะนำในการเพิ่มช่องที่เก็บเมาส์	63
5.6.1.2 ควรมีน้ำหนักที่เบาและสามารถเคลื่อนย้ายได้สะดวก.....	63
5.6.1.3 ข้อเสนอแนะทางเทคนิค.....	63
5.6.1.4 ตัวเครื่องควรมีระบบรักษาความปลอดภัยมากกว่านี้ เช่น กุญแจล็อก ช่องที่เก็บ Notebook ไว้	63
บรรณานุกรม	64
ภาคผนวก ก แบบเสนอร่างโครงการ	65
ภาคผนวก ข ความคืบหน้าโครงการ	73
ภาคผนวก ค คู่มือการใช้งาน	91
ภาคผนวก ง ประวัติผู้จัดทำ.....	93

สารบัญตาราง

ตารางที่	หน้า
1.5 ระยะเวลาการทำโครงการ	4
3.1 แสดงแผนการดำเนินโครงการ	53

สารบัญรูป

รูปที่	หน้า
1.1 แผนภาพ Flowchart ขั้นตอนการดำเนินงาน	3
2.1 การต่อวงจรควบคุมมอเตอร์ DC หมุนซ้าย หมุนขวา.....	11
2.2 แสดงผลชุดคำสั่งมอเตอร์	12
2.3 การใช้คำสั่ง if ลงบอร์ด Arduino.....	15
2.4 การใช้คำสั่ง การใช้คำสั่ง if...else ลงบอร์ด Arduino	16
2.5 สัญลักษณ์ ฟังก์ชัน	34
2.6 บอร์ด Arduino.....	40
2.7 รูปการเขียนโปรแกรมบนArduino	40
2.8 เลือกบอร์ด Arduino ที่ต้องการ Upload	43
2.9 เลือกหมายเลข Comport ของบอร์ด.....	43
2.10 Layout & Pin out Arduino Board.....	44
2.11 ข้อมูลของส่วนต่างๆในบอร์ด Arduino	44
2.12 วงจรสวิตช์ NodeMCU v.10.....	47
2.13 การอัปโหลดโค้ดลง NodeMCU	50
3.1 แสดง Flowchart รูปแบบการทำงานของชุดคำสั่ง	55
4.1 การประกาศตัวแปร และการ Setup IN/OUT.....	60
4.2 การทำงานของปุ่มกดในรูปแบบ loop	61
4.3 การทำงานของมอเตอร์ (เปิดช่องโน้ตบุ๊ก)	61
4.4 การทำงานของมอเตอร์ (ปิดช่องโน้ตบุ๊ก).....	62

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันมีโต๊ะมากมายจนนับไม่ถ้วน บอกได้ว่าโต๊ะมีความสำคัญต่อการใช้ชีวิตประจำวันของใครหลายคน (อ้างอิงจาก www.moveworldtogether.com จากบทความ โต๊ะตัวเล็กกับชีวิตที่ดีขึ้น) และโต๊ะนั้นยังสามารถแยกได้หลายแบบอีกด้วย ขึ้นอยู่กับการใช้งาน เช่น โต๊ะสำหรับกินข้าว โต๊ะสำหรับทำงาน โต๊ะสำหรับเรียนหนังสือ ทำให้การดำรงชีวิตของเรามีความสะดวกสบายขึ้นเป็นอย่างมาก หากโต๊ะไม่ได้ถูกสร้างขึ้นมาก็จะทำให้เกิดผลเสียต่าง ๆ ตามมา เช่น การทำงานที่ลำบากไม่สะดวกสบาย หรือ อาจจะเสียสุขภาพได้เนื่องจากการที่ทำงานไม่ถูกท่า โดยปกติจะนั่งหลังตรงหากไม่มีโต๊ะก็ต้องก้มซึ่งจะทำให้หลังค่อมได้ และแน่นอนว่าโต๊ะนั้นต้องแข็งแรงปลอดภัย คงทนต่อการใช้งานต่าง ๆ หรือ การใช้งานโดยเฉพาะ เพื่อให้ผู้ใช้สามารถใช้งานได้อย่างสบายใจ

ทางคณะผู้จัดทำจึงได้เห็นว่าโต๊ะนั้นมีความสำคัญจึงคิดที่จะสร้างโต๊ะขึ้น ซึ่งได้ยกปัญหาที่เกิดขึ้นในการใช้งานโต๊ะ ก็คือเรื่องพื้นที่ในการทำงานที่ไม่เพียงพอ ซึ่งอาจจะเป็นปัญหาที่เกิดขึ้นกับใครบางคน และทางคณะผู้จัดทำจึงคิดที่จะสร้างโต๊ะญี่ปุ่นที่มีขนาดไม่ใหญ่มากเหมาะกับการใช้งานที่บ้าน ที่ทำงาน หรือ ที่คอนโดมิเนียม โดยปัญหาที่จะต้องแก้ไขคือเรื่องพื้นที่ในการทำงานที่มีน้อย คณะผู้จัดทำจึงได้เจาะจงไปที่กลุ่มคนที่ใช้ Notebook เพราะหากวาง Notebook ก็เต็มพื้นที่แล้ว ทำให้ไม่สามารถมีพื้นที่ในการทำงาน คณะผู้จัดทำจึงคิดที่จะทำโต๊ะที่สามารถเก็บ Notebook ได้ โดยนำเทคโนโลยีเข้ามามีส่วนช่วยในการทำงานของโต๊ะ เพื่อให้โต๊ะสามารถใช้งานได้อย่างคุ้มค่า และทันสมัย สะดวกต่อผู้ใช้งาน และมีความปลอดภัยต่อ Notebook โดยโต๊ะที่สร้างจะทำจากไม้อัด มีความกว้าง 1 เมตร ยาว 20 นิ้ว สูง 16 นิ้ว รองรับ Notebook ทุกขนาดตั้งแต่ 15.6 – 21 นิ้ว และ สร้างช่องเก็บที่เปิด-ปิด Notebook เป็นระบบอัตโนมัติมีการทำงานโดยการกดปุ่ม เพื่อเปิดใช้งาน มีการควบคุมผ่านชุดคำสั่งจากบอร์ด Arduino UNO R3 เพื่อสั่งการมอเตอร์ในการบังคับช่องเก็บ Notebook ให้สามารถเปิด-ปิดได้แบบอัตโนมัติ

คณะผู้จัดทำจึงเริ่มจัดทำโครงการ โต๊ะเก็บโน้ตบุ๊กอัตโนมัติ ที่ไว้ใช้เก็บ Notebook ขึ้นเพื่อแก้ไขปัญหาพื้นที่การทำงานที่จำกัด และช่วยให้ Notebook มีความปลอดภัยไม่เสียหาย

1.2 วัตถุประสงค์ของโครงการ

1.2.1 เพื่อสร้างโต๊ะสำหรับเก็บโน้ตบุ๊กใช้งานในพื้นที่จำกัด

1.2.2 เพื่อจัดเก็บ Notebook ได้อย่างปลอดภัย

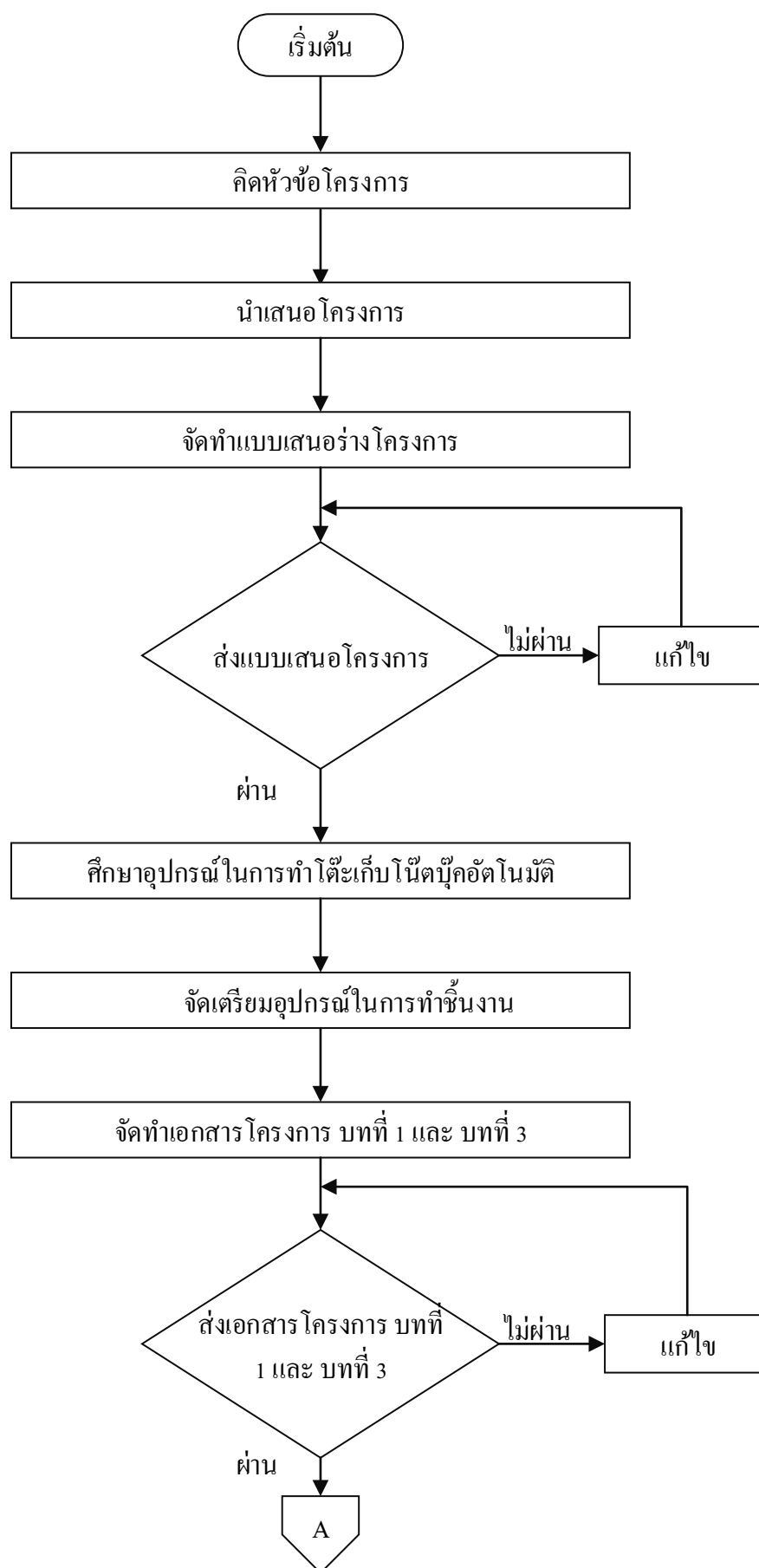
1.2.3 เพื่อนำความรู้ที่ได้จากการเรียนการสอนในสาขาวิชาเทคโนโลยีสารสนเทศมา
ประยุกต์ใช้ในการทำโครงการ

1.3 ขอบเขตของโครงการ

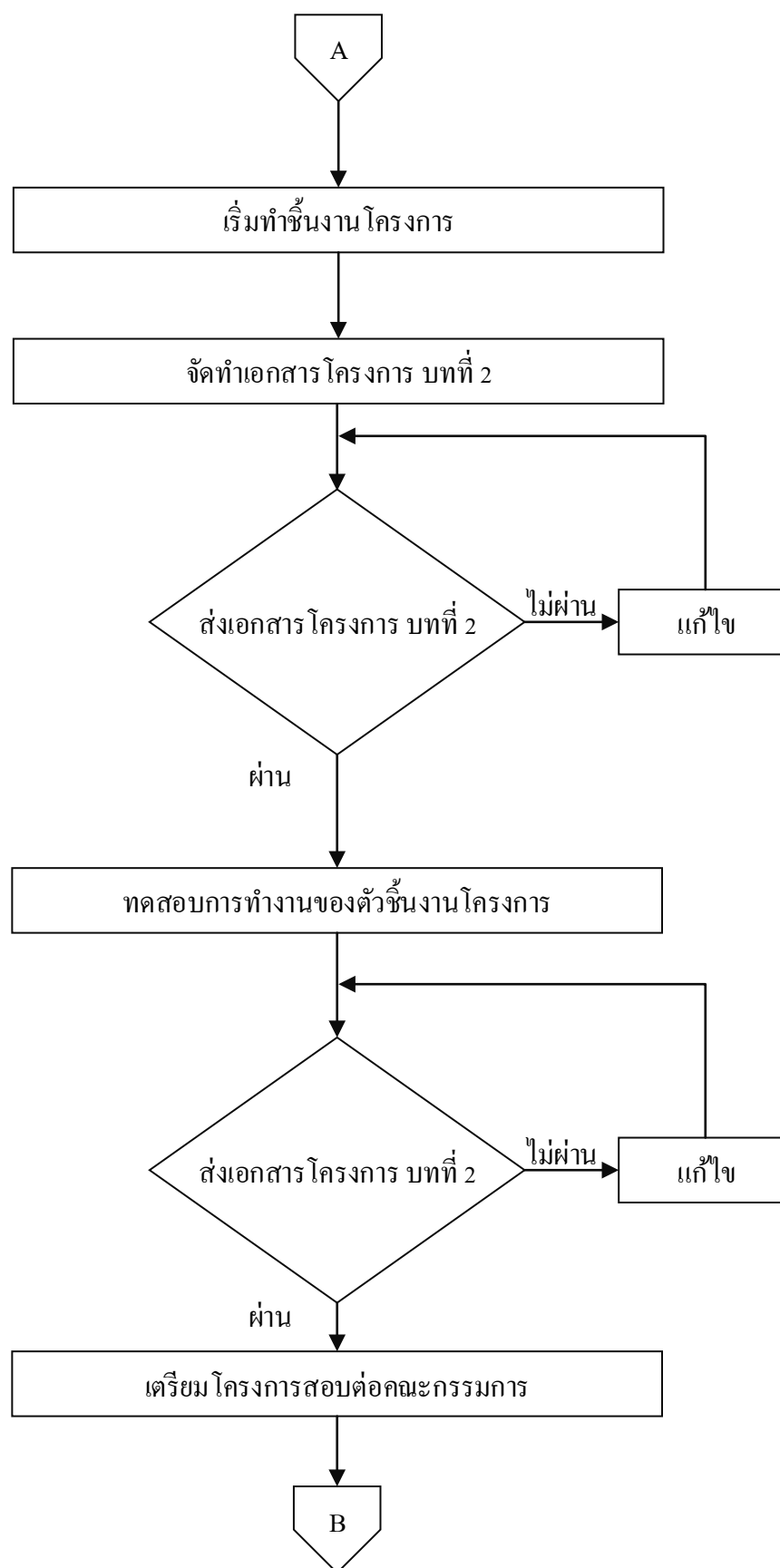
- 1.3.1 ใช้ภาษา C ในการเขียนโปรแกรม
- 1.3.2 ใช้โปรแกรม Arduino 1.8.9 ในการเขียนชุดคำสั่ง
- 1.3.3 ใช้คำสั่งการทำงานแบบเงื่อนไข if...else
- 1.3.4 มีชุดคำสั่งควบคุมมอเตอร์ให้ทำงาน
- 1.3.5 ชุดคำสั่งจะทำงานแบบ Loop หากครบเงื่อนไขจะกลับไปทำซ้ำ

1.4 ขั้นตอนการดำเนินงาน

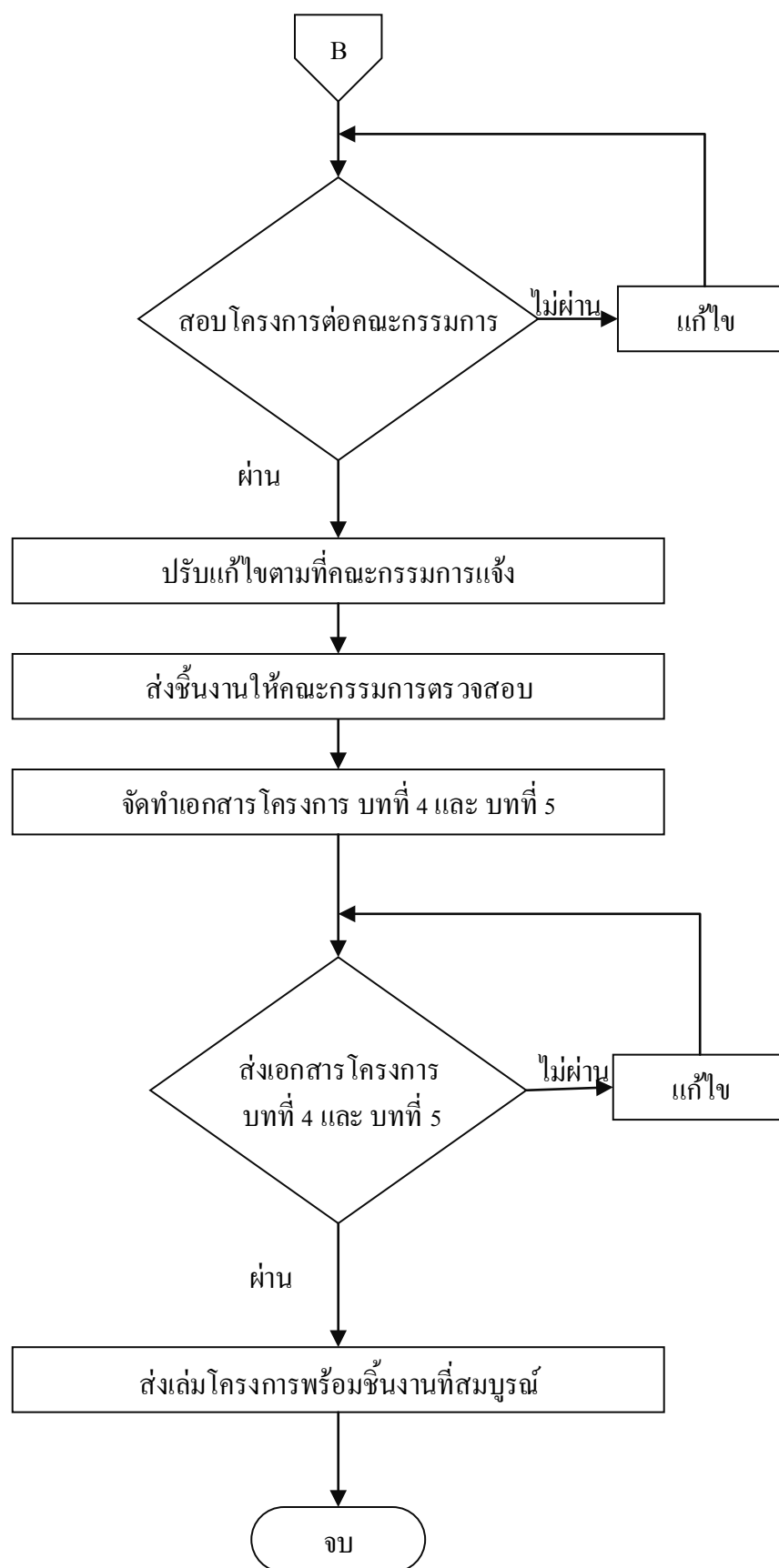
- 1.4.1 คิดหัวข้อโครงการ
- 1.4.2 นำเสนอโครงการ
- 1.4.3 จัดทำแบบเสนอร่างโครงการ
- 1.4.4 ส่งแบบเสนอโครงการ
- 1.4.5 ศึกษาอุปกรณ์ในการทำโต๊ะเก็บโน้ตบุ๊กอัตโนมัติ
- 1.4.6 จัดเตรียมอุปกรณ์ในการทำชิ้นงาน
- 1.4.7 จัดทำเอกสารโครงการ บทที่ 1 และ บทที่ 3
- 1.4.8 ส่งเอกสารโครงการ บทที่ 1 และ บทที่ 3
- 1.4.9 เริ่มทำชิ้นงานโครงการ
- 1.4.10 จัดทำเอกสารโครงการ บทที่ 2
- 1.4.11 ส่งเอกสารโครงการ บทที่ 2
- 1.4.12 ทดสอบการทำงานของตัวชิ้นงานโครงการ
- 1.4.13 แก้ไขข้อบกพร่องที่เกิดขึ้นในการทดสอบ
- 1.4.14 เตรียมโครงการสอบต่อคณะกรรมการ
- 1.4.15 สอบโครงการต่อคณะกรรมการ
- 1.4.16 แก้ไขตามที่คณะกรรมการแจ้ง
- 1.4.17 ส่งชิ้นงานให้คณะกรรมการตรวจสอบ
- 1.4.18 จัดทำเอกสารโครงการ บทที่ 4-5
- 1.4.19 ส่งเอกสารโครงการ บทที่ 4-5
- 1.4.20 ส่งเล่มโครงการพร้อมชิ้นงานที่สมบูรณ์



รูปที่ 1.1 แผนภาพ Flowchart ขั้นตอนการดำเนินงาน



รูปที่ 1.1 แผนภาพ Flowchart ขั้นตอนการดำเนินงาน (ต่อ)



รูปที่ 1.1 แผนภาพ Flowchart ขั้นตอนการดำเนินงาน (ต่อ)

1.6 ประโยชน์ที่คาดว่าจะได้รับ

- 1.6.1 ได้โต๊ะสำหรับเก็บโน้ตบุ๊กใช้งานในพื้นที่จำกัด
- 1.6.2 จัดเก็บ Notebook ได้อย่างปลอดภัย
- 1.6.3 ได้นำความรู้ที่ได้จากการเรียนการสอนในสาขาวิชาเทคโนโลยีสารสนเทศมาประยุกต์ใช้ในการทำโครงการ

1.7 งบประมาณที่ใช้ในการทำโครงการ

1.7.1	Arduino UNO R3	260	บาท
1.7.2	L298N motor driver	180	บาท
1.7.3	Switching 12 V 10 A	390	บาท
1.7.4	Jump Wire (Male to Female)	120	บาท
1.7.5	ปลั๊กตัวผู้	13	บาท
1.7.6	มอเตอร์เกียร์ DC 12V	1,600	บาท
1.7.7	สายไฟ	30	บาท
1.7.8	ไม้อัด	500	บาท
1.7.9	ปุ่มกด	35	บาท
1.7.10	เฟืองสะพาน	800	บาท
1.7.11	เฟืองกลม	225	บาท
1.7.12	เล่มโครงการ	200	บาท
1.7.13	ซีดี CD	50	บาท
1.7.14	อื่น ๆ	1,000	บาท
	ราคารวม	<u>5,403</u>	บาท

บทที่ 2

ความรู้เบื้องต้นและทฤษฎีที่เกี่ยวข้อง

ความรู้และทฤษฎีที่เกี่ยวข้องกับโครงการชุดคำสั่งควบคุมมอเตอร์ไต่เก็บโน้ตบุ๊กอิเล็กทรอนิกส์ ซึ่งมีความรู้มากและ ทฤษฎี ที่นำมาใช้ในการสร้างโครงการในครั้งนี้ เพื่อให้โครงการสำเร็จ

มีความสมบูรณ์ และ ประสพผลสำเร็จ

คณะผู้จัดทำโครงการจึงได้นำความรู้ที่เกี่ยวข้องมาใช้กับกับโครงการ ดังนี้

- 2.1 สอนใช้งาน Arduino ควบคุมมอเตอร์ DC หมุนซ้าย หมุนขวา
- 2.2 สอนใช้งาน Arduino if else ตรวจสอบเงื่อนไข ถูกผิด
- 2.3 ภาษาซี (C Programming Language)
- 2.4 ผังงาน (Flowchart)
- 2.5 โปรแกรม Arduino
- 2.6 การเขียนโปรแกรมคอมพิวเตอร์ (Computer programming)
- 2.7 หลักการทำงานของบอร์ด Arduino

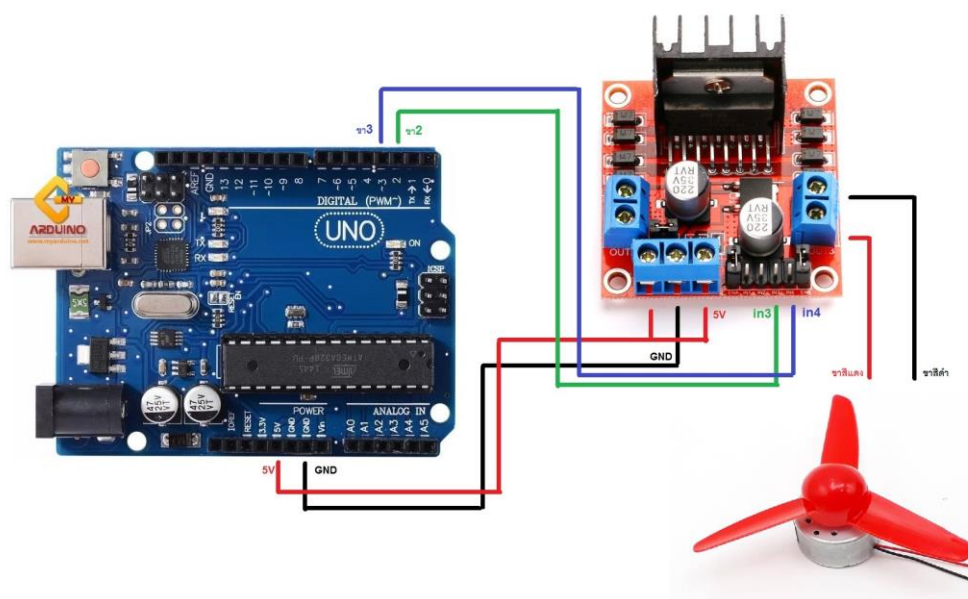
2.1 สอนใช้งาน Arduino ควบคุมมอเตอร์ DC หมุนซ้าย หมุนขวา

บทความนี้จะสอนใช้งาน Arduino ควบคุม มอเตอร์หมุนซ้ายหมุนขวา โดยใช้ตัวขับเคลื่อนมอเตอร์ L298N การทำให้มอเตอร์หมุนซ้ายและหมุนขวากลับทางได้นั้น เราต้องสลับขั้วการจ่ายไฟ โดยใช้ โมดูล L298N ควบคุมการจ่ายไฟสลับทางให้กับมอเตอร์

อุปกรณ์ที่ต้องใช้ใน บทความ สอนใช้งาน Arduino ควบคุมมอเตอร์ DC หมุนซ้าย หมุนขวา

- Arduino UNO R3 แดมฟรี สายUSB
- บอร์ดทดลอง Breadboard 830 Point
- สายไฟจัมเปอร์ ผู้-ผู้ ยาว 20cm. จำนวน 40 เส้น
- สายไฟจัมเปอร์ ผู้-เมีย ยาว 20cm. จำนวน 40 เส้น
- L298N Motor Drive Module
- ใบพัดมอเตอร์ 80mm สีแดง
- DC Motor มอเตอร์ แนวตั้ง 3-6 Volt
- วิธีการต่ออุปกรณ์ สอนใช้งาน Arduino ควบคุมมอเตอร์ DC หมุนซ้าย หมุนขวา
- Arduino uno r3 -> โมดูลขับเคลื่อนมอเตอร์ L298N
- 5V -> 5V

- GND -> GND
- 5V -> +12V
- ขา2 -> In3
- ขา3 -> In4
- โมดูลขับเคลื่อนมอเตอร์ L298N -> Motor
- OUT3 -> ขาสีแดง
- OUT4 -> ขาสีดำ



รูปที่ 2.1 การต่อวงจรควบคุมมอเตอร์ DC หมุนซ้าย หมุนขวา

(www.myarduino.net/article/101/สอนใช้งาน-arduino-ควบคุมมอเตอร์-dc-หมุนซ้าย-หมุนขวา)

```
int MotorPin2 = 2;
int MotorPin3 = 3;

void setup() {
  pinMode(MotorPin2, OUTPUT);
  pinMode(MotorPin3, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  digitalWrite(MotorPin2, HIGH); //สั่งงานให้โมดูลขับเคลื่อนมอเตอร์จ่ายไฟ ออกขา OUT3
  digitalWrite(MotorPin3, LOW); //สั่งงานให้ขา OUT4 เป็นขารับไฟจากขา OUT3
  Serial.println("Motor Right");
```

```

delay(2000);
digitalWrite(MotorPin2, LOW); //สั่งงานให้ขา OUT3 หยุดจ่ายไฟ
digitalWrite(MotorPin3, LOW); //สั่งงานให้ขา OUT4 หยุดจ่ายไฟ
Serial.println("Motor STOP");
delay(2000);
digitalWrite(MotorPin2, LOW); //สั่งงานให้ขา OUT3 เป็นขารับไฟจากขา OUT4
digitalWrite(MotorPin3, HIGH); //สั่งงานให้โมดูลขับเคลื่อนมอเตอร์จ่ายไฟ ออกขา OUT4
Serial.println("Motor Left");
delay(2000);
}

```

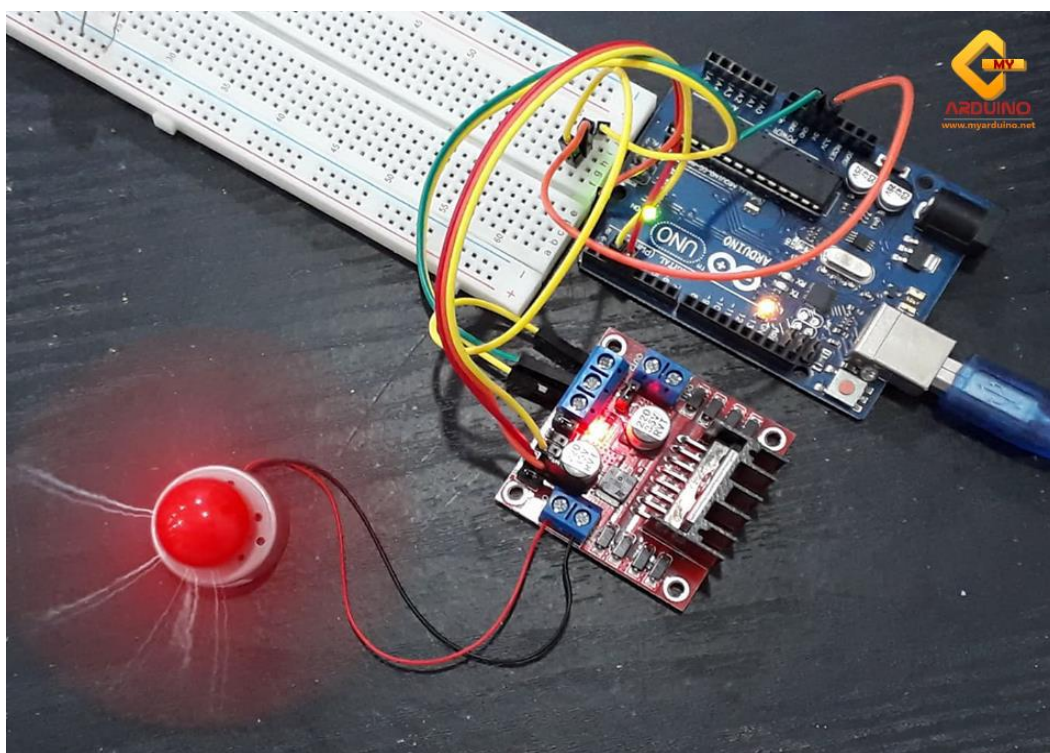
จากโค้ดตัวอย่าง จะทำการควบคุม โมดูลขับเคลื่อนมอเตอร์ L298N จ่ายไฟให้ มอเตอร์ โปรแกรมจะสั่งให้ โมดูลขับเคลื่อนมอเตอร์ จ่ายไฟออกขา OUT3 และ ให้ไฟไหลกลับเข้า OUT4 ทำให้มอเตอร์หมุน

```
digitalWrite(MotorPin2, HIGH);
```

```
digitalWrite(MotorPin3, LOW);
```

แล้ว สั่งงานให้ หยุดจ่ายไฟขา OUT3 OUT4 ทำให้มอเตอร์หยุด

แล้ว สั่งงานให้จ่ายไฟออกขา OUT4 และ ให้ไฟไหลกลับเข้า OUT3 ทำให้มอเตอร์หมุนกลับทาง ตามลำดับ



รูปที่ 2.2 แสดงผลชุดคำสั่งมอเตอร์

2.2 ส่วนใช้งาน Arduino if else ตรวจสอบเงื่อนไข ถูกผิด

คำสั่ง if เป็นคำสั่งใช้สำหรับตรวจสอบเงื่อนไข เพื่อสั่งให้โปรแกรมเลือกทำงานในปีกกาต่างๆ ตามผลลัพธ์ที่ได้จากการตรวจสอบเงื่อนไขของคำสั่ง โดยมีรูป 2 แบบดังนี้

รูปแบบที่ 1 เป็นแบบ If

```
if (เงื่อนไข) {
```

คำสั่งที่ต้องการให้ทำเมื่อเงื่อนไขเป็นจริง

```
}
```

ตัวอย่างคำสั่ง if

```
int i = 1;
```

```
if (i == 1) {
```

ถ้า i เท่ากับ 1 จะทำในปีกกา

```
}
```

รูปแบบที่ 2 เป็นแบบ If...else

```
if (เงื่อนไข) {
```

คำสั่งที่ต้องการให้ทำเมื่อเงื่อนไขเป็นจริง

```
}
```

```
else {
```

คำสั่งที่ต้องการให้ทำเมื่อเงื่อนไขเป็นจริงเท็จ

```
}
```

ตัวอย่างคำสั่ง If...else

```
int x = 1 ;
```

```
if (x == 1) {
```

ถ้า x เท่ากับ 1 จะทำในปีกกา ถ้าไม่ใช่จะข้ามไปทำในปีกกา Else

```
}
```

```
else {
```

ถ้า X ไม่เท่ากับ 1 จะทำในปีกกา

```
}
```

คำสั่งที่ใช้ตรวจสอบเงื่อนไข ได้แก่

- == หมายความว่า เท่ากับ
- != หมายความว่า ไม่เท่ากับ

- < หมายความว่า น้อยกว่า
- > หมายความว่า มากกว่า
- <= หมายความว่า น้อยกว่าหรือเท่ากับ
- >= หมายความว่า มากกว่าหรือเท่ากับ
- && หมายความว่า และ
- || หมายความว่า หรือ

ให้ก๊อปโค้ดข้างล่างทดสอบ การใช้คำสั่ง if ลงบอร์ด Arduino

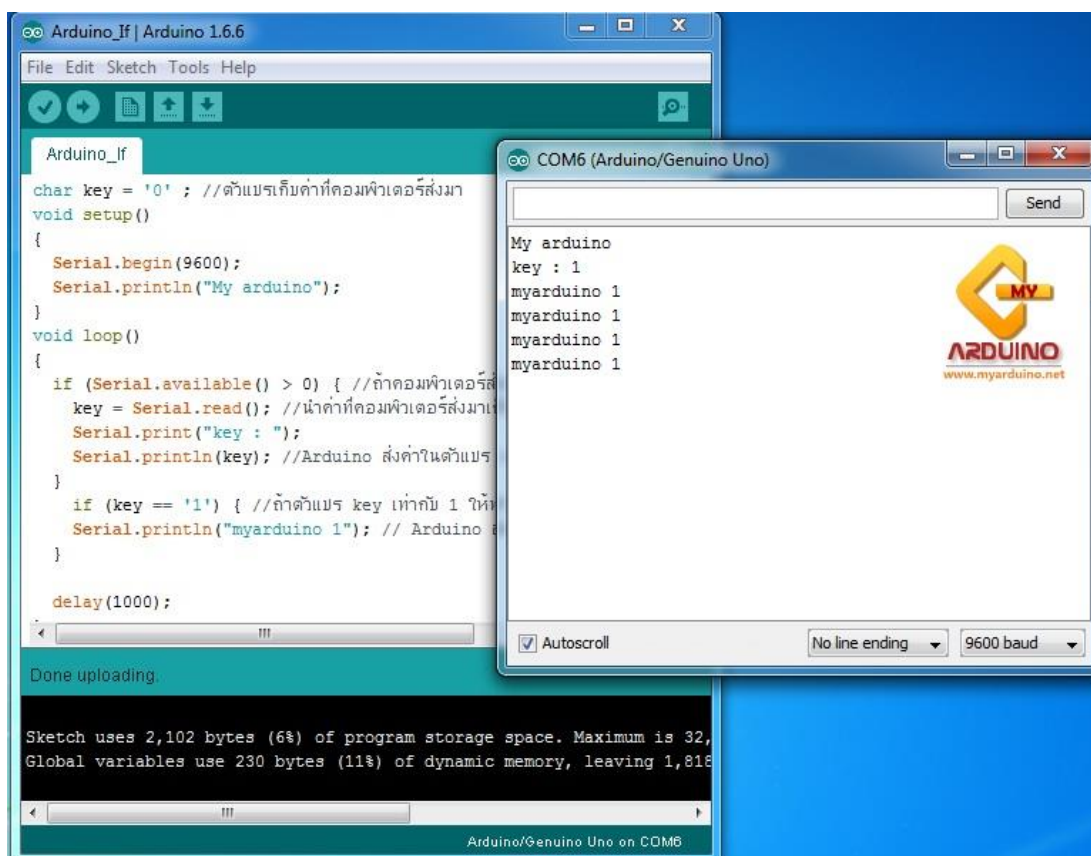
char key = '0'; //ตัวแปรเก็บค่าที่คอมพิวเตอร์ส่งมา

void setup()

```
{
  Serial.begin(9600);
  Serial.println("My arduino");
}
```

void loop()

```
{
  if (Serial.available() > 0) { //ถ้าคอมพิวเตอร์ส่งข้อมูลมาจะทำใน if นี้
    key = Serial.read(); //นำค่าที่คอมพิวเตอร์ส่งมาเก็บในตัวแปร key
    Serial.print("key : ");
    Serial.println(key); //Arduino ส่งค่าในตัวแปร key เข้าคอมพิวเตอร์ Serial Monitor
  }
  if (key == '1') { //ถ้าตัวแปร key เท่ากับ 1 ให้ทำในปีกกานี้
    Serial.println("myarduino 1"); // Arduino ส่งข้อความตอบกลับมาทาง
    Serial Monitor "myarduino 1"
  }
  delay(1000);
}
```



รูปที่ 2.3 การใช้คำสั่ง if ลงบอร์ด Arduino

(www.myarduino.net/article/87/สอนใช้งาน-arduino-if-else-ตรวจสอบเงื่อนไข-ถูกผิด)

ให้ก๊อปปี้โค้ดข้างล่างทดสอบ การใช้คำสั่ง if...else ลงบอร์ด Arduino

char key = '0'; //ตัวแปรเก็บค่าที่คอมพิวเตอร์ส่งมา

void setup()

{

Serial.begin(9600);

Serial.println("My arduino");

}

void loop()

{

if (Serial.available() > 0) { //ถ้าคอมพิวเตอร์ส่งข้อมูลมาเราจะทำใน if นี้

key = Serial.read(); //นำค่าที่คอมพิวเตอร์ส่งมาเก็บในตัวแปร key

Serial.print("key : ");

Serial.println(key); //Arduino ส่งค่าในตัวแปร key เข้าคอมพิวเตอร์ Serial Monitor

}

```

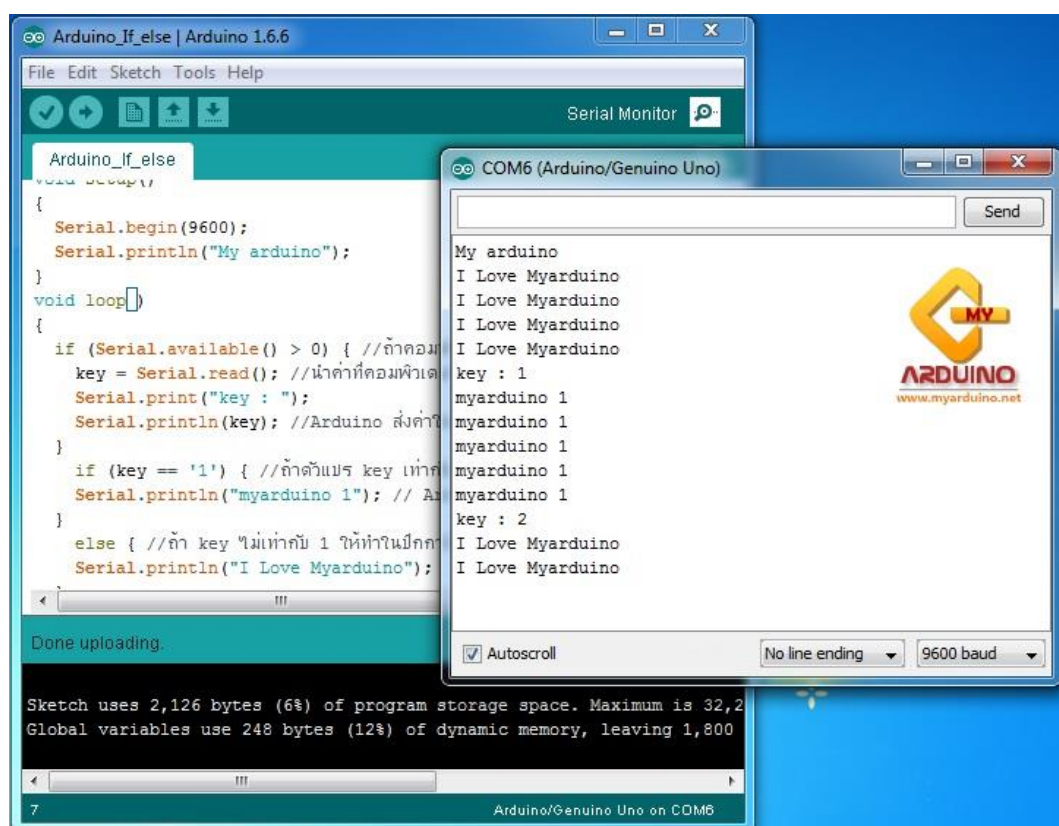
if (key == '1') { //ถ้าตัวแปร key เท่ากับ 1 ให้ทำในปีกกานี้
  Serial.println("myarduino 1"); // Arduino ส่งข้อความตอบกลับมาทาง Serial
  Monitor "myarduino 1"
}

else { //ถ้า key ไม่เท่ากับ 1 ให้ทำในปีกกานี้
  Serial.println("I Love Myarduino");
}

delay(1000);
}

```

โค้ดข้างบนเป็นการใช้คำสั่ง if..else ในการตรวจสอบเงื่อนไข ให้พิมพ์ 1 แล้วกด Send ส่งค่า 1 จากคอมพิวเตอร์มาเก็บไว้ในตัวแปร key แล้วให้ Arduino ตรวจสอบค่า key ถ้าตัวแปร key เท่ากับ 1 ให้ Arduino ส่งตอบกลับมาทาง Serial Monitor ว่า myarduino 1 ตามรูปด้านล่าง ต่อมาให้ลอง พิมพ์ 2 แล้วกด Send ถ้าค่า key ไม่เท่ากับ 1 ให้ทำในปีกกา else ให้ Arduino ส่งตอบกลับมาทาง Serial Monitor ว่า I Love Myarduino ตามรูปด้านล่าง



รูปที่ 2.4 การใช้คำสั่ง การใช้คำสั่ง if...else ลงบอร์ด Arduino

2.3 ภาษาซี (C Programming Language)

ภาษาซี (C) เป็นภาษาโปรแกรมสำหรับวัตถุประสงค์ทั่วไป เริ่มพัฒนาขึ้นระหว่าง พ.ศ. 2512–2516 (ค.ศ. 1969–1973) โดยเดนนิส ริตชี (Dennis Ritchie) ที่เอทีแอนด์ทีเบลล์แล็บส์ (AT&T Bell Labs) ภาษาซีเป็นภาษาที่มีความยืดหยุ่นในการเขียนโปรแกรมและมีเครื่องมืออำนวยความสะดวกสำหรับการเขียนโปรแกรมเชิงโครงสร้างและอนุญาตให้มีขอบข่ายตัวแปร (scope) และการเรียกซ้ำ (recursion) ในขณะที่ระบบชนิดตัวแปรอพลวัตก็ช่วยป้องกันการดำเนินการที่ไม่ตั้งใจหลายอย่าง เหมือนกับภาษาโปรแกรมเชิงคำสั่งส่วนใหญ่ในแบบแผนของภาษาอัลกอล การออกแบบของภาษาซีมีคอนสตรัคต์ (construct) ที่โยงกับชุดคำสั่งเครื่องทั่วไปได้อย่างพอเพียง จึงทำให้ยังมีการใช้ในโปรแกรมประยุกต์ซึ่งแต่ก่อนลงรหัสเป็นภาษาแอสเซมบลี คือซอฟต์แวร์ระบบอันโดดเด่นอย่างระบบปฏิบัติการคอมพิวเตอร์ ยูนิกซ์

ภาษาซีเป็นภาษาโปรแกรมหนึ่งที่ใช้กันอย่างแพร่หลายมากที่สุดตลอดกาล และตัวแปลโปรแกรมของภาษาซีมิให้ใช้งานได้สำหรับสถาปัตยกรรมคอมพิวเตอร์และระบบปฏิบัติการต่าง ๆ เป็นส่วนมาก

ภาษาหลายภาษาในยุคหลังได้หยิบยืมภาษาซีไปใช้ทั้งทางตรงและทางอ้อม ตัวอย่างเช่น ภาษาดี ภาษาโก ภาษารัสต์ ภาษาจาวา จาวาสคริปต์ ภาษาลิมโบ ภาษาแอลพีซี ภาษาซีชาร์ป ภาษาอ็อบเจกทีฟ-ซี ภาษาเพิร์ล ภาษาพีเอชพี ภาษาไพทอน ภาษาเวโรลีโอก (ภาษาพรรณาสารค์แวร์) และซีเชลล์ของยูนิกซ์ ภาษาเหล่านี้ได้ดึงโครงสร้างการควบคุมและคุณลักษณะพื้นฐานอื่น ๆ มาจากภาษาซี ส่วนใหญ่มีวากยสัมพันธ์คล้ายคลึงกับภาษาซีเป็นอย่างมากโดยรวม (ยกเว้นภาษาไพทอนที่ต่างออกไปอย่างสิ้นเชิง) และตั้งใจที่จะผสมผสานนิพจน์และข้อความสั่งที่จำแนกได้ของวากยสัมพันธ์ของภาษาซี ด้วยระบบชนิดตัวแปร ตัวแบบข้อมูล และอรรถศาสตร์ที่อาจแตกต่างกันโดยมูลฐาน ภาษาซีพลัสพลัสและภาษาอ็อบเจกทีฟ-ซีเดิมเกิดขึ้นในฐานะตัวแปลโปรแกรมที่สร้างรหัสภาษาซี ปัจจุบันภาษาซีพลัสพลัสแทบจะเป็นเซตใหญ่ของภาษาซี^[9] ในขณะที่ภาษาอ็อบเจกทีฟ-ซีก็เป็นเซตใหญ่อันเคร่งครัดของภาษาซี

ก่อนที่จะมีมาตรฐานภาษาซีอย่างเป็นทางการ ผู้ใช้และผู้พัฒนาต่างก็เชื่อถือในข้อกำหนดอย่างไม่เป็นทางการในหนังสือที่เขียนโดยเดนนิส ริตชี และไบรอัน เคอร์นิกัน (Brian Kernighan) ภาษาซีรุ่นนั้นจึงเรียกกันโดยทั่วไปว่า *ภาษาเคแอนด์อาร์ซี* (K&R C) ต่อมา พ.ศ. 2532 สถาบันมาตรฐานแห่งชาติของสหรัฐอเมริกา (ANSI) ได้ตีพิมพ์มาตรฐานสำหรับภาษาซีขึ้นมา เรียกกันว่า *ภาษาแอนซีซี* (ANSI C) หรือ *ภาษาซี89* (C89) ในปีถัดมา องค์การระหว่างประเทศว่าด้วยการมาตรฐาน (ISO) ได้อนุมัติให้ข้อกำหนดเดียวกันนี้เป็นมาตรฐานสากล เรียกกันว่า *ภาษาซี90* (C90) ในเวลาต่อมาอีก องค์การฯ ก็ได้เผยแพร่ส่วนขยายมาตรฐานเพื่อรองรับสากลวิวัตน์ (internationalization) เมื่อ พ.ศ. 2538 และมาตรฐานที่ตรวจชำระใหม่เมื่อ พ.ศ. 2542 เรียกกัน

ว่า ภาษาซี99 (C99) มาตรฐานรุ่นปัจจุบันก็ได้รับอนุมัติเมื่อเดือนธันวาคม พ.ศ. 2554 เรียกกันว่า ภาษาซี12 (C12)

ภาษาซีเป็นภาษาที่ใช้ในการมีปฏิสัมพันธ์เช่น เชิงคำสั่ง (หรือเชิงกระบวนการ) ถูกออกแบบขึ้นเพื่อใช้แปลด้วยตัวแปลโปรแกรมแบบการเชื่อมโยงที่ตรงไปตรงมา สามารถเข้าถึงหน่วยความจำในระดับล่าง เพื่อสร้างภาษาที่จับคู่อย่างมีประสิทธิภาพกับชุดคำสั่งเครื่อง และแทบไม่ต้องการสนับสนุนใด ๆ ขณะทำงาน ภาษาซีจึงเป็นประโยชน์สำหรับหลายโปรแกรมที่ก่อนหน้านี้เคยเขียนในภาษาแอสเซมบลีมาก่อน

หากคำนึงถึงความสามารถในระดับล่าง ภาษาซีถูกออกแบบขึ้นเพื่อส่งเสริมการเขียนโปรแกรมที่ขึ้นอยู่กับเครื่องใดเครื่องหนึ่ง (machine-independent) โปรแกรมภาษาซีที่เขียนขึ้นตามมาตรฐานและเคลื่อนย้ายได้ สามารถแปลได้บนแพลตฟอร์มคอมพิวเตอร์และระบบปฏิบัติการต่าง ๆ อย่างกว้างขวาง โดยแก้ไขรหัสต้นฉบับเพียงเล็กน้อยหรือไม่ต้องแก้ไขเลย ภาษาซีสามารถใช้ได้บนแพลตฟอร์มได้หลากหลายตั้งแต่ไมโครคอนโทรลเลอร์ฝังตัวไปจนถึง[[แอสรีเคน))

ภาษาซีมีสิ่งอำนวยความสะดวกสำหรับการเขียนโปรแกรมเชิงโครงสร้าง และสามารถกำหนดขอบข่ายตัวแปรและเรียกซ้ำ เช่นเดียวกับภาษาโปรแกรมเชิงคำสั่งส่วนใหญ่ในสายตระกูลภาษาอัลกอล ในขณะที่ระบบชนิดตัวแปรแบบอพลวัตช่วยป้องกันการดำเนินการที่ไม่ได้ตั้งใจ รหัสที่ทำงานได้ทั้งหมดในภาษาซีถูกบรรจุอยู่ในฟังก์ชัน พารามิเตอร์ของฟังก์ชันส่งผ่านด้วยค่าของตัวแปรเสมอ ส่วนการส่งผ่านด้วยการอ้างอิงจะถูกจำลองขึ้นโดยการส่งผ่านค่าตัวชี้ ชนิดข้อมูลรวมแบบแตกต่าง (struct) ช่วยให้สมาชิกข้อมูลที่เกี่ยวข้องกันสามารถรวมกันและจัดการได้ในหน่วยเดียว รหัสต้นฉบับของภาษาซีเป็นรูปแบบอิสระ ซึ่งใช้อัฒภาค (;) เป็นตัวจบคำสั่ง (มิใช่ตัวแบ่ง)

ภาษาซียังมีลักษณะเฉพาะต่อไปนี้เพิ่มเติม

- ตัวแปรอาจถูกซ่อนในบล็อกซ้อนใน
- ชนิดตัวแปรไม่เคร่งครัด เช่นข้อมูลตัวอักษรสามารถใช้เป็นจำนวนเต็ม
- เข้าถึงหน่วยความจำคอมพิวเตอร์ในระดับต่ำโดยแปลงที่อยู่ในเครื่องด้วยชนิดตัวแปรตัวชี้ (pointer)
- ฟังก์ชันและตัวชี้ข้อมูลรองรับการทำงานในภาวะหลายรูปแบบ (polymorphism)
- การกำหนดดัชนีแถวลำดับสามารถทำได้ด้วยวิธีรอง คือนิยามในพจน์ของเลขคณิตของตัวชี้
- ตัวประมวลผลก่อนสำหรับการนิยามแมโคร การรวมไฟล์รหัสต้นฉบับ และการแปลโปรแกรมแบบมีเงื่อนไข
- ความสามารถที่ซับซ้อนเช่น ไอ/โอ การจัดการสายอักขระ และฟังก์ชันทางคณิตศาสตร์รวมอยู่ในไลบรารี
- คำหลักที่สงวนไว้มีจำนวนค่อนข้างน้อย

- ตัวดำเนินการแบบประสมจำนวนมาก เช่น `+=`, `-=`, `*=`, `++` ฯลฯ
โครงสร้างการเขียน คล้ายภาษาบีมากกว่าภาษาอัลกอล ตัวอย่างเช่น
- ใช้วงเล็บปีกกา `{ ... }` แทนที่จะเป็น `begin ... end` ในภาษาอัลกอล 60 หรือวงเล็บโค้ง `(...)` ในภาษาอัลกอล 68
- เท่ากับ `=` ใช้สำหรับกำหนดค่า (คัดลอกข้อมูล) เหมือนภาษาฟอร์แทรน แทนที่จะเป็น `:=` ในภาษาอัลกอล
- เท่ากับสองตัว `==` ใช้สำหรับเปรียบเทียบความเท่ากัน แทนที่จะเป็น `.EQ.` ในภาษาฟอร์แทรนหรือ `=` ในภาษาเบสิกและภาษาอัลกอล
- ตรรกะ "และ" กับ "หรือ" แทนด้วย `&&` กับ `||` ตามลำดับ แทนที่จะเป็นตัวดำเนินการ `^` กับ `V` ในภาษาอัลกอล แต่ตัวดำเนินการดังกล่าวจะไม่ประเมินค่าตัวถูกดำเนินการทางขวา ถ้าหากผลลัพธ์จากทางซ้ายสามารถพิจารณาได้แล้ว เหตุการณ์เช่นนี้เรียกว่าการประเมินค่าแบบลัดวงจร (short-circuit evaluation) และตัวดำเนินการดังกล่าวก็มีความหมายต่างจากตัวดำเนินการระดับบิต `&` กับ `|`

ธรรมชาติของภาษาในระดับต่ำช่วยให้โปรแกรมเมอร์ควบคุมสิ่งที่คอมพิวเตอร์กระทำได้อย่างใกล้ชิด ในขณะที่อนุญาตให้มีการปรับแต่งพิเศษและการทำให้เหมาะสมที่สุดสำหรับแพลตฟอร์มหนึ่งใดโดยเฉพาะ สิ่งนี้ทำให้รหัสสามารถทำงานได้อย่างมีประสิทธิภาพบนฮาร์ดแวร์ที่มีทรัพยากรจำกัดมาก ๆ ได้เช่นระบบฝังตัว

ภาษาซีไม่มีคุณลักษณะบางอย่างที่มีในภาษาอื่นอาทิ

- ไม่มีการนิยามฟังก์ชันซ้อนใน
- ไม่มีการกำหนดค่าแถวลำดับหรือสายอักขระโดยตรง (การคัดลอกข้อมูลจะกระทำผ่านฟังก์ชันมาตรฐาน แต่รองรับการกำหนดค่าวัตถุที่มีชนิดเป็น `struct` หรือ `union`)
- ไม่มีการเก็บข้อมูลขยะโดยอัตโนมัติ
- ไม่มีข้อกำหนดเพื่อการตรวจสอบขอบเขตของแถวลำดับ
- ไม่มีการดำเนินการสำหรับแถวลำดับทั้งหมดในระดับตัวภาษา
- ไม่มีวากยสัมพันธ์สำหรับช่วงค่า (range) เช่น `A..B` ที่ใช้ในบางภาษา
- ก่อนถึงภาษาซี99 ไม่มีการแบ่งแยกชนิดข้อมูลแบบบูล (ค่าศูนย์หรือไม่ศูนย์ถูกนำมาใช้แทน)^[12]
- ไม่มีส่วนปิดคลุมแบบรูปนัย (closure) หรือฟังก์ชันในรูปแบบพารามิเตอร์ (มีเพียงตัวชี้ของฟังก์ชันและตัวแปร)

- ไม่มีตัวสร้างและโครูทีน การควบคุมกระแสการทำงานภายในเทร็ดมีเพียงการเรียกใช้ฟังก์ชันซ้อนลงไป เว้นแต่การใช้ฟังก์ชัน `longjmp` หรือ `setcontext` จากไลบรารี
- ไม่มีการจัดการสิ่งที่ผิดปกติ (exception handling) ฟังก์ชันไลบรารีมาตรฐานจะแสดงเงื่อนไขข้อผิดพลาดด้วยตัวแปรส่วนกลาง `errno` และ/หรือค่ากลับคืนพิเศษ และฟังก์ชันไลบรารีได้เตรียม `goto` แบบไม่ใช่เฉพาะที่ไว้ด้วย
- การเขียนโปรแกรมเชิงมอดูลรองรับแค่ระดับพื้นฐานเท่านั้น
- การโอเวอร์โหลดฟังก์ชันหรือตัวดำเนินการไม่รองรับภาวะหลายรูปแบบขณะแปลโปรแกรม
- การเขียนโปรแกรมเชิงวัตถุรองรับในระดับที่จำกัดมาก โดยพิจารณาจากภาวะหลายรูปแบบกับการรับทอด (inheritance)
- การซ่อนสารสนเทศ (encapsulation) รองรับในระดับที่จำกัด
- ไม่รองรับโดยพื้นฐานกับการทำงานแบบมัลติเทร็ดและเครือข่ายคอมพิวเตอร์
- ไม่มีไลบรารีมาตรฐานสำหรับคอมพิวเตอร์กราฟิกส์และความจำเป็นหลายอย่างในการเขียนโปรแกรมประยุกต์

คุณลักษณะเหล่านี้จำนวนหนึ่งมิให้ใช้ได้จากส่วนขยายในตัวแปลโปรแกรมบางตัว หรือจัดสรรไว้แล้วในสภาพแวดล้อมของระบบปฏิบัติการ (เช่น โปสซิกซ์) หรือจัดเตรียมโดยไลบรารีภายนอก หรือสามารถจำลองโดยดัดแปลงแก้ไขรหัสที่มีอยู่ หรือบางครั้งก็ถูกพิจารณาว่าไม่ใช่รูปแบบการเขียนโปรแกรมที่เหมาะสม

การดำเนินการหลายอย่างในภาษาซีมีพฤติกรรมไม่นิยามซึ่งไม่ถูกกำหนดว่าต้องตรวจสอบขณะแปลโปรแกรม ในกรณีของภาษาซี "พฤติกรรมไม่นิยาม" หมายถึงพฤติกรรมเฉพาะอย่างที่เกิดขึ้นโดยมาตรฐานมิได้ระบุไว้ และสิ่งที่จะเกิดขึ้นก็ไม่มีในเอกสารการใช้งานของภาษาซี หนึ่งในชุดคำสั่งที่มีชื่อเสียงและน่าขบขันจากกลุ่มข่าว `comp.std.c` และ `comp.lang.c` นั้นทำให้โปรแกรมเกิดปัญหาที่เรียกว่า "ปีศาจที่ออกมาจากจมูกของคุณ" (demons to fly out of your nose) บางครั้งสิ่งที่เกิดขึ้นในทางปฏิบัติอันเป็นผลมาจากพฤติกรรมไม่นิยามทำให้เกิดจุดบกพร่องที่ยากต่อการตรวจสอบและอาจทำให้ข้อมูลในหน่วยความจำผิดแปลกไป ตัวแปลโปรแกรมบางชนิดช่วยสร้างการดำเนินงานที่ทำให้พฤติกรรมนั้นดีขึ้นและมีเหตุผล ซึ่งแตกต่างจากการแปลโดยตัวแปลชนิดอื่นที่อาจดำเนินงานไม่เหมือนกัน สาเหตุที่พฤติกรรมบางอย่างยังคงไว้ว่าไม่นิยามก็เพื่อให้ตัวแปลโปรแกรมบนสถาปัตยกรรมชุดของคำสั่งเครื่องที่หลากหลาย สามารถสร้างรหัสที่ทำงานได้ในพฤติกรรมที่นิยามอย่างมีประสิทธิภาพมากขึ้น ซึ่งเชื่อว่าเป็นบทบาทหนึ่งที่สำคัญของภาษาซีในฐานะภาษาสำหรับสร้างระบบ ดังนั้นภาษาซีจึงส่งผลให้เกิดความรับผิดชอบของโปรแกรมเมอร์เพื่อ

หลีกเลี่ยงพฤติกรรมไม่นิยาม โดยอาจใช้เครื่องมือต่าง ๆ เพื่อค้นหาส่วนของโปรแกรมว่าพฤติกรรมใดบ้างที่ไม่นิยาม ตัวอย่างของพฤติกรรมไม่นิยามเช่น

- การเข้าถึงข้อมูลนอกขอบเขตของแวลวลำดับ
- ข้อมูลล้น (overflow) ในตัวแปรจำนวนเต็มมีเครื่องหมาย
- ฟังก์ชันที่กำหนดไว้ว่าต้องส่งค่ากลับ แต่ไม่มีคำสั่งส่งกลับ (return) ในฟังก์ชัน ในขณะที่เดียวกันคำสั่งกลับก็ถูกใช้งานด้วย
- การอ่านค่าตัวแปรโดยที่ยังไม่ได้กำหนดค่าเริ่มต้น

การดำเนินการเหล่านี้ทั้งหมดเป็นข้อผิดพลาดในการเขียนโปรแกรม ซึ่งสามารถปรากฏในการใช้ภาษาโปรแกรมอื่น ๆ จำนวนมาก ภาษาซีจึงถูกวิพากษ์วิจารณ์เพราะมาตรฐานของมันสามารถชี้ให้เห็นถึงพฤติกรรมไม่นิยามในหลายกรณีได้อย่างชัดเจน รวมไปถึงพฤติกรรมบางอย่างที่อาจนิยามไว้อย่างดีแล้ว และไม่มีการระบุกลไกการจัดการกระทำต่อข้อผิดพลาดขณะทำงานเลย

ตัวอย่างหนึ่งของพฤติกรรมไม่นิยามเช่นการเรียกใช้ `flush()` บนกระแสนข้อมูลป้อนเข้า ซึ่งไม่จำเป็นว่าจะทำให้โปรแกรมทำงานผิดพลาด แต่ในบางกรณีที่ทำให้เกิดผลที่สอดคล้องกันได้นิยามไว้แล้วอย่างดี มีความหมายซึ่งใช้ประโยชน์ได้ (จากตัวอย่างนี้คือการสมมติให้ข้อมูลที่ป้อนเข้าถูกละทิ้งทั้งหมดจนถึงอักขระขึ้นบรรทัดใหม่ตัวถัดไป) เป็น *ส่วนขยาย* ที่อนุญาต ส่วนขยายที่ไม่เป็นมาตรฐานเช่นนี้เป็นข้อจำกัดความสามารถในการเคลื่อนย้ายของซอฟต์แวร์

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

จุดเริ่มต้นของภาษาซีผูกอยู่กับการพัฒนาระบบปฏิบัติการยูนิกซ์อย่างใกล้ชิด ซึ่งเดิมพัฒนาด้วยภาษาแอสเซมบลีบนหน่วยประมวลผลพีดีพี-7 โดยริตชีและทอมป์สัน โดยผสมผสานความคิดหลากหลายจากเพื่อนร่วมงาน ในตอนท้ายพวกเขาคัดสินใจที่จะย้ายระบบปฏิบัติการนั้นลงในพีดีพี-11 แต่ภาษาบีขาดความสามารถบางอย่างที่จะใช้คุณลักษณะอันได้เปรียบของพีดีพี-11 เช่นความสามารถในการระบุตำแหน่งที่อยู่เป็นไบต์ จึงทำให้เกิดการพัฒนาภาษาซีรุ่นแรกขึ้นมา

รุ่นดั้งเดิมของระบบยูนิกซ์บนพีดีพี-11ถูกพัฒนาขึ้นด้วยภาษาแอสเซมบลี เมื่อประมาณ พ.ศ. 2516 ภาษาซีเพิ่มชนิดข้อมูล `struct` ทำให้ภาษาซีเพียงพออย่างมีประสิทธิภาพ ซึ่งเคอร์เนलयูนิกซ์ส่วนใหญ่ถูกเขียนด้วยภาษาซี นี่ก็เป็นเคอร์เนลหนึ่งของระบบปฏิบัติการที่พัฒนาด้วยภาษาอื่นนอกเหนือจากภาษาแอสเซมบลี (ระบบอื่นเช่นมัลดิกส์เขียนด้วยภาษาพีแอล/วัน เอ็มซีพีสำหรับเบอร์โรส์ ปี5000เขียนด้วยภาษาอัลกอล ในปี พ.ศ. 2504)

เมื่อ พ.ศ. 2521 ไบรอัน เคอร์นิกัน (Brian Kernighan) และเคนนิส ริดจ์ ได้ตีพิมพ์หนังสือเล่มแรกชื่อ *เดอะซีโปรแกรมมิงแลงกวิจ (The C Programming Language)* ซึ่งเป็นที่รู้จักในกลุ่มโปรแกรมเมอร์ภาษาซีว่า "เคแอนด์อาร์" (K&R อักษรย่อของผู้แต่งทั้งสอง) หนังสือเล่มนี้ทำหน้าที่เป็นข้อกำหนดของภาษาอย่างไม่เป็นทางการมาหลายปี ภาษาซีรุ่นดังกล่าวจึงมักถูกอ้างถึงว่าเป็น ภาษาเคแอนด์อาร์ซี (K&R C) ส่วนหนังสือที่ปรับปรุงครั้งที่สองครอบคลุมมาตรฐานแอนซีซีที่มีขึ้นทีหลัง

ภาษาเคแอนด์อาร์ซีได้แนะนำคุณลักษณะหลายประการเช่น

- ไลบรารีไอ/โอมาตรฐาน
- ชนิดข้อมูล `long int` (จำนวนเต็มขนาดยาว)
- ชนิดข้อมูล `unsigned int` (จำนวนเต็มไม่มีเครื่องหมาย)
- ตัวดำเนินการกำหนดค่าแบบประสมในรูปแบบ `=ตัวดำเนินการ` (เช่น `==`) ถูกเปลี่ยนเป็น `ตัวดำเนินการ=` (เช่น `--`) เพื่อลดปัญหาความกำกวมเชิงความหมาย อย่างเช่นกรณี `i=-10` ซึ่งจะถูกตีความว่า `i = -10` แทนที่จะเป็นอย่างที่ตั้งใจคือ `i = -i`

แม้ว่าหลังจากการเผยแพร่มาตรฐานของภาษาซีเมื่อ พ.ศ. 2532 ภาษาเคแอนด์อาร์ซีถูกพิจารณาว่าเป็น "ส่วนร่วมต่ำสุด" อยู่เป็นเวลาหลายปี (ความสามารถในการแปลรหัสจำนวนหนึ่งเป็นคำสั่งซึ่งทำงานได้บนเครื่องใดก็ตามเป็นอย่างน้อย) ซึ่งโปรแกรมเมอร์ภาษาซีต้องจำกัดความสามารถของพวกเขาในกรณีที่ต้องการให้ระบบสามารถใช้ได้กับหลายเครื่องมากที่สุด เนื่องจากตัวแปลโปรแกรมเก่า ๆ ก็ยังคงมีการใช้งานอยู่ และการเขียนภาษาซีแบบเคแอนด์อาร์อย่างระมัดระวังสามารถเข้ากันได้กับภาษาซีมาตรฐานเป็นอย่างดี

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 สามารถละเว้นได้ในภาษาเคแอลเอ็นอาร์ซี แต่ `long int` จำเป็นต้องประกาศ

การประกาศฟังก์ชันของภาษาเคแอลเอ็นอาร์ซีไม่มีการระบุข้อมูลเกี่ยวกับอาร์กิวเมนต์ที่ใช้ ดังนั้นจึงไม่มีการตรวจชนิดข้อมูลพารามิเตอร์ของฟังก์ชัน แม้ว่าตัวแปลโปรแกรมบางตัวจะแสดงข้อความเตือน ถ้าฟังก์ชันถูกเรียกใช้ภายในโดยมีจำนวนอาร์กิวเมนต์ที่ผิด หรือถ้าฟังก์ชันถูกเรียกใช้หลายครั้งจากภายนอกโดยมีชนิดข้อมูลของอาร์กิวเมนต์ต่างกัน เครื่องมือภายนอกอาทิ ลินต์ (lint) ของยูนิกซ์ถูกพัฒนาขึ้นเพื่อให้สามารถตรวจสอบความคงเส้นคงวาของฟังก์ชันที่ใช้งานข้ามไฟล์รหัสต้นฉบับหลายไฟล์

หลายปีถัดจากการเผยแพร่ภาษาเคแอลเอ็นอาร์ซี คุณลักษณะที่ไม่เป็นทางการหลายอย่างก็ถูกเพิ่มเข้ามาในภาษา ซึ่งรองรับโดยตัวแปลโปรแกรมจากเอทีแอนด์ทีและผู้ผลิตรายอื่น คุณลักษณะที่เพิ่มเหล่านี้เช่น

- ฟังก์ชัน `void`
- ฟังก์ชันที่คืนค่าเป็นชนิดข้อมูล `struct` หรือ `union` (แทนที่จะเป็นตัวชี้)
- การกำหนดค่าให้กับชนิดข้อมูล `struct`
- ชนิดข้อมูลแฉงนับ (enumerated type)

ส่วนขยายที่เพิ่มขึ้นอย่างมากและการขาดข้อตกลงในเรื่องไลบรารีมาตรฐาน อีกทั้งความนิยมในภาษาและข้อเท็จจริงที่ว่าไม่เพียงแต่ตัวแปลโปรแกรมยูนิกซ์เท่านั้นที่พัฒนาขึ้นตามข้อกำหนดของเคแอลเอ็นอาร์ซี ทั้งหมดนำไปสู่ความสำคัญของการทำให้เป็นมาตรฐาน

การเขียนโปรแกรมระบบเป็นการใช้งานหลักของภาษาซี ซึ่งรวมไปถึงการพัฒนา ระบบปฏิบัติการและโปรแกรมประยุกต์ระบบฝังตัว เนื่องจากลักษณะเฉพาะอันเป็นที่ต้องการถูกรวมเข้าไว้ด้วยกัน อย่างเช่น ความสามารถในการเคลื่อนย้ายได้กับประสิทธิภาพของรหัสต้นฉบับ ความสามารถในการเข้าถึงที่อยู่ของฮาร์ดแวร์ที่ระบุ ความสามารถเรื่อง type punning เพื่อให้เข้ากับ

ความต้องการการเข้าถึงข้อมูลที่กำหนดไว้จากภายนอก และความต้องการทรัพยากรระบบขณะทำงานต่ำ ภาษาซีสามารถใช้เขียนโปรแกรมเว็บไซต์โดยใช้ซีจีไอเป็น "เกตเวย์" เพื่อแลกเปลี่ยนสารสนเทศระหว่างเว็บแอปพลิเคชัน เซิร์ฟเวอร์ และเบราว์เซอร์ ปัจจุบันอย่างที่ทำให้เลือกภาษาซีแทนที่จะเป็นภาษาอินเทอร์พรีเตอร์ คือความเร็ว เสถียรภาพ และความอ่อนไหวต่อการเปลี่ยนแปลงในสภาพแวดล้อมของการดำเนินงาน เนื่องจากเป็นธรรมชาติของภาษาคอมไพเลอร์

ผลจากการยอมรับในระดับกว้างขวางและประสิทธิภาพของภาษาซี ทำให้ตัวแปลโปรแกรมตัวแปลคำสั่ง ไบรารีต่าง ๆ ของภาษาอื่น มักพัฒนาขึ้นด้วยภาษาซี ตัวอย่างเช่น ตัวแปลโปรแกรมภาษาไอเฟลหลายโปรแกรมส่งข้อมูลออกเป็นรหัสภาษาซีเป็นภาษากลาง เพื่อส่งต่อไปให้ตัวแปลโปรแกรมภาษาซีต่อไป การพัฒนาสายหลักของภาษาไพทอน ภาษาเพิร์ล 5 และภาษาพีเอชพี ทั้งหมดถูกเขียนขึ้นด้วยภาษาซี

ภาษาซีมีประสิทธิภาพสำหรับคอมพิวเตอร์เพื่องานคำนวณและวิทยาศาสตร์ เนื่องจากความสิ้นเปลืองต่ำ ธรรมชาติของภาษาระดับต่ำ ธรรมชาติของภาษาที่ถูกแปล และมีส่วนคณิตศาสตร์ที่ดีในไลบรารีมาตรฐาน ตัวอย่างของการใช้ภาษาซีในงานคำนวณและวิทยาศาสตร์ เช่น จีเอ็มพี ไลบรารีวิทยาศาสตร์ของกนู แมเทอแมติกา แมตแล็บ และแซส

ภาษาซีบางครั้งใช้เป็นภาษาระหว่างกลางในการทำให้เกิดผลของภาษาอื่น แนวคิดนี้อาจใช้เพื่อความสะดวกต่อการเคลื่อนย้าย โดยให้ภาษาซีเป็นภาษาระหว่างกลาง ซึ่งไม่จำเป็นต้องพัฒนาตัวสร้างรหัสแบบเจาะจงเครื่อง ตัวแปลโปรแกรมที่ใช้ภาษาซีในทางนี้เช่น บิตซี แกมบิต จีเอชซี สควิก และวาลา เป็นต้น อย่างไรก็ตามภาษาซีถูกออกแบบมาเพื่อเป็นภาษาเขียนโปรแกรม ไม่ใช่ภาษาเป้าหมายของตัวแปลโปรแกรม จึงเหมาะสมน้อยกว่าสำหรับการใช้เป็นภาษาระหว่างกลางด้วยเหตุผลนี้นำไปสู่การพัฒนาภาษาระหว่างกลางที่มีพื้นฐานบนภาษาซีเช่น ภาษาซีโมนัสไมนัส

ผู้ใช้งานปลายใช้ภาษาซีอย่างแพร่หลายเพื่อสร้างแอปพลิเคชันของผู้ใช้เอง แต่เมื่อแอปพลิเคชันใหญ่ขึ้น การพัฒนาเช่นนั้นมักจะย้ายไปทำในภาษาอื่นที่พัฒนามาด้วยกัน เช่น ภาษาซีพลัสพลัส ภาษาซีชาร์ป ภาษาวิชวลเบสิก เป็นต้น

ภาษาซีรองรับตัวดำเนินการหลายประเภท ซึ่งเป็นสัญลักษณ์ที่ใช้ในนิพจน์เพื่อระบุการจัดการที่จะถูกทำให้เกิดผล ระหว่างการประเมินค่าของนิพจน์นั้น ภาษาซีมีตัวดำเนินการต่อไปนี้

- พิชคณิต (+, -, *, /, %)
- การกำหนดค่า (=)
- การกำหนดค่าแต่งเดิม (+=, -=, *=, /=, %=, &=, |=, ^=, <<=, >>=)
- ตรรกะระดับบิต (~, &, |, ^)
- การเลื่อนระดับบิต (<<, >>)
- ตรรกะแบบบูล (!, &&, ||)

- การประเมินค่าเชิงเงื่อนไข (?:)
- การทดสอบภาวะเท่ากัน (==, !=)
- การรวมอาร์กิวเมนต์ฟังก์ชัน (())
- การเพิ่มค่าและการลดค่า (++ , --)
- การเลือกสมาชิกในวัตถุ (., ->)
- ขนาดของวัตถุ (sizeof)
- ความสัมพันธ์เชิงอันดับ (<, <=, >, >=)
- การอ้างอิงและการถูกอ้างอิง (&, *, [])
- การลำดับ (,)
- การจัดกลุ่มนิพจน์ย่อย (())
- การแปลงชนิดข้อมูล (())

ภาษาซีมีไวยากรณ์รูปนิพจน์ซึ่งระบุโดยมาตรฐานภาษาซี

วากยสัมพันธ์ของการแปลงชนิดข้อมูลสามารถใช้แปลงค่าต่าง ๆ ระหว่างชนิดข้อมูลจำนวนเต็มและจำนวนจุดลอยตัว (จำนวนทศนิยม) หรือระหว่างจำนวนเต็มสองจำนวน หรือระหว่างจำนวนจุดลอยตัวสองจำนวน ที่มีขนาดแตกต่างกัน ตัวอย่างเช่น `(long int)sqrt(1000.0)`, `(double)(256*256)` หรือ `(float)sqrt(1000.0)` เป็นต้น การแปลงชนิดข้อมูลเป็นภาวะปริยายในหลายบริบท อาทิ เมื่อกำหนดค่าให้กับตัวแปรหรือพารามิเตอร์ของฟังก์ชัน หรือเมื่อใช้จำนวนจุดลอยตัวเป็นดัชนีของเวกเตอร์ หรือในการดำเนินการทางเลขคณิตที่มีตัวถูกดำเนินการเป็นข้อมูลคนละชนิดกัน

การแปลงค่าระหว่างจำนวนเต็มและจำนวนจุดลอยตัวโดยทั่วไป จะเกิดการเปลี่ยนแปลงการเข้ารหัสระดับบิตไปยังขอบเขตที่เป็นไปได้เพื่อสงวนค่าจำนวนของตัวถูกดำเนินการนั้น ไม่เหมือนกับการแปลงชนิดข้อมูลกรณีอื่น (ซึ่งการเข้ารหัสระดับบิตของตัวถูกดำเนินการจะถูกตีความใหม่ตามชนิดเป้าหมายเพียงเท่านั้น) โดยเฉพาะอย่างยิ่ง การแปลงชนิดข้อมูลจากจำนวนเต็มไปเป็นจำนวนจุดลอยตัวจะคงไว้ซึ่งค่าจำนวนได้อย่างถูกต้อง เว้นแต่ถ้าจำนวนบิตในชนิดเป้าหมายมีไม่เพียงพอ กรณีดังกล่าวจะทำให้บิตที่มีนัยสำคัญน้อยที่สุดสูญหายไป

ส่วนการแปลงชนิดข้อมูลจากจำนวนจุดลอยตัวไปเป็นจำนวนเต็มจะเกิดการตัดค่าหลังจุดทศนิยมอย่างหลีกเลี่ยงไม่ได้ (ค่าถูกปัดเศษเข้าหาศูนย์) สำหรับการปัดเศษชนิดอื่น ภาษาซี 99 ได้ระบุไว้แล้วในฟังก์ชันดังนี้ (ใน `<math.h>`)

- `round()`: ปัดเศษไปยังจำนวนเต็มที่ใกล้ที่สุด

- `rint()`, `nearbyint()`: ปัดเศษตามทิศทางของจำนวนจุดลอยตัวปัจจุบัน
- `ceil()`: ค่าจำนวนเต็มน้อยสุดที่ไม่น้อยกว่าอาร์กิวเมนต์ (ปัดขึ้น) ดูเพิ่มที่ฟังก์ชันเพดาน
- `floor()`: ค่าจำนวนเต็มมากสุดที่ไม่มากกว่าอาร์กิวเมนต์ (ปัดลง) ดูเพิ่มที่ฟังก์ชันพื้น
- `trunc()`: ปัดเศษเข้าหาศูนย์ (เหมือนกับการแปลงชนิดข้อมูลเป็นจำนวนเต็ม)

ฟังก์ชันทั้งหมดนี้รับอาร์กิวเมนต์ `double` และคืนค่าเป็น `double` ซึ่งต่อจากนี้ก็อาจแปลงชนิดข้อมูลเป็นจำนวนเต็มอีกทีหากจำเป็น

การแปลงชนิดข้อมูลจาก `float` ไปเป็น `double` จะคงไว้ซึ่งค่าจำนวนได้อย่างถูกต้อง ในขณะที่การแปลงกลับ ค่าจะถูกปัดเศษซึ่งมักเป็นการปัดเศษเข้าหาศูนย์เพื่อให้พอดีกับจำนวนบิตที่น้อยลง (เนื่องจาก `float` ก็มีช่วงเลขชี้กำลังที่น้อยกว่าด้วย การแปลงชนิดข้อมูลอาจให้ผลเป็นค่าอนันต์แทน) ตัวแปลโปรแกรมบางโปรแกรมจะแปลงค่าของ `float` ไปเป็น `double` โดยเบื้องหลัง ในบางบริบทเช่น พารามิเตอร์ของฟังก์ชันที่ประกาศเป็น `float` ตามความเป็นจริงอาจส่งค่าเป็น `double` ก็ได้

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

ภาษาซีมีระบบชนิดตัวแปรแบบไม่เคร่งครัด ซึ่งมีความคล้ายคลึงบางประการร่วมกับภาษาลูกของภาษาอัลกอล อาทิ ภาษาปาสคาล ภาษาซีมีชนิดตัวแปรที่เตรียมไว้แล้วสำหรับจำนวนเต็มหลายขนาด แบบทั้งมีเครื่องหมายและไม่มีเครื่องหมาย จำนวนจุดลอยตัว ตัวอักษร และชนิดข้อมูลแฉก (enum) ในภาษาซี99 ได้เพิ่มชนิดตัวแปรแบบบูลเข้าไปด้วย ภาษาซีก็ยังมีชนิดตัวแปรที่รับทอดมาด้วยเช่นแถวลำดับ ตัวชี้ ระเบียบ (struct) และยูเนียน (union)

ภาษาซีมักใช้กับการเขียนโปรแกรมระบบในระดับต่ำ ซึ่งอาจหลีกเลี่ยงการใช้ระบบชนิดตัวแปรเมื่อจำเป็น ตัวแปลโปรแกรมจะพยายามทำให้แน่ใจว่า ชนิดตัวแปรถูกใช้อย่างถูกต้องในนิพจน์ส่วนใหญ่ แต่โปรแกรมเมอร์ก็สามารถลบการตรวจสอบเช่นนั้นได้หลายทาง อาทิ การโยนชนิดข้อมูล (type cast) เพื่อแปลงค่าจากชนิดหนึ่งไปเป็นชนิดหนึ่งอย่างชัดเจน หรือการใช้ตัวชี้หรือยูเนียนเพื่อแปลความหมายบิตของค่าที่อยู่ภายในไปเป็นอีกชนิดหนึ่ง

ภาษาซีรองรับการใช้งานตัวชี้ (pointer) ซึ่งเป็นชนิดข้อมูลสำหรับการอ้างอิงอย่างง่ายชนิดหนึ่ง ที่เก็บบันทึกที่อยู่หรือตำแหน่งของวัตถุหรือฟังก์ชันในหน่วยความจำ ตัวชี้สามารถ อ้างอิงกลับ (dereference) เพื่อเข้าถึงข้อมูลที่บันทึกในตำแหน่งที่ถูกชี้อยู่ หรือเพื่อเรียกใช้ฟังก์ชันที่ถูกชี้อยู่ ตัวชี้สามารถจัดดำเนินการกำหนดค่าและเลขคณิตของตัวชี้ได้ด้วย ค่าของตัวชี้ขณะ โปรแกรมทำงาน

มักจะเป็นตำแหน่งมูลฐานในหน่วยความจำ (ซึ่งอาจเสริมด้วยค่าออฟเซตในหน่วยเวิร์ด) แต่เนื่องจากตัวชี้มีการระบุชนิดตามข้อมูลที่ชี้ไป ตัวแปลโปรแกรมจึงสามารถตรวจสอบชนิดตัวแปรในนิพจน์ต่าง ๆ รวมทั้งตัวชี้ด้วยกันเองขณะแปลได้ เลขคณิตของตัวชี้จะแปรสัดส่วนของขนาดโดยอัตโนมัติตามชนิดข้อมูลที่ชี้ไป (ดูเพิ่มที่ส่วนความเข้าใจแทนกันได้ระหว่างตัวชี้และแถวลำดับ) จุดประสงค์ของการใช้ตัวชี้มีหลากหลายในภาษาซีเช่น สายอักขระมักจัดดำเนินการโดยใช้ตัวชี้ไปยังแถวลำดับของตัวอักขระ การจัดสรรหน่วยความจำพลวัต (dynamic memory allocation) สามารถกระทำได้ด้วยตัวชี้ ชนิดข้อมูลชนิดอื่นเช่น ต้นไม้ ปกติจะถูกพัฒนาขึ้นโดยการจัดสรรวัตถุ `struct` โดยพลวัต ซึ่งเชื่อมโยงแต่ละหน่วยเข้ากันด้วยตัวชี้ ตัวชี้ของฟังก์ชันใช้เพื่อการเรียกกลับ (callback) สำหรับชุดคำสั่งจัดการเหตุการณ์ เป็นต้น

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

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

การใช้งานตัวชี้อย่างไม่ระมัดระวังอาจเกิดอันตรายได้ เนื่องจากตัวแปรตัวชี้สามารถชี้ไปที่ตำแหน่งใดก็ได้โดยไม่มีการกั้นเขต และปกติก็ไม่มีการตรวจสอบ ซึ่งอาจทำให้เกิดผลกระทบที่ไม่พึงปรารถนา ถึงแม้ตัวชี้ที่ใช้งานอย่างถูกต้องได้ชี้ไปยังตำแหน่งที่ปลอดภัยอยู่แล้ว แต่มันก็อาจถูกทำให้ชี้ไปยังตำแหน่งที่ไม่ปลอดภัยโดยการดำเนินการเลขคณิตที่ไม่ถูกต้อง หรือตัวชี้ไปยังวัตถุที่อาจเรียกคืนการจัดสรรไปแล้วแต่ถูกเรียกใช้ใหม่ (ตัวชี้อย่างหลวม dangling pointer) หรือตัวชี้ที่อาจใช้งานโดยไม่กำหนดค่าเริ่มต้น (ตัวชี้ตัวแทน wild pointer) หรือตัวชี้ที่อาจถูกกำหนดด้วยค่าที่ไม่ปลอดภัยโดยตรง ด้วยวิธีโฮนชนิดตัวแปร ยูเนียน หรือผ่านค่ามาจากตัวชี้อื่นที่เสีย เป็นต้น โดยทั่วไปภาษาซีอนุญาตให้จัดดำเนินการและแปลงชนิดตัวแปรของตัวชี้ได้ แม้ว่าตัวแปลโปรแกรมก็มีตัวเลือกสำหรับการตรวจสอบอยู่หลายระดับก็ตาม ภาษาโปรแกรมอื่นบางภาษาจัดการปัญหานี้โดยกำหนดให้ใช้ชนิดตัวแปรอ้างอิงที่เคร่งครัดมากกว่า

ชนิดข้อมูลแถวลำดับ (array) ในภาษาซีแบบดั้งเดิมมีขนาดคงที่และสถิต ซึ่งจะถูกกำหนดตอนแปลโปรแกรม (ในเวลาถัดมา มาตรฐานภาษาซี99 อนุญาตให้สร้างแถวลำดับที่มีความยาวแปรได้) อย่างไรก็ตามแถวลำดับสามารถกำหนดให้จัดสรรเนื้อที่หน่วยความจำขนาดใดก็ได้ขณะทำงาน

โดยใช้ฟังก์ชัน `malloc` จากไลบรารีมาตรฐาน แล้วทำให้เป็นแถวลำดับ การทำให้เป็นหนึ่งเดียวระหว่างแถวลำดับและตัวชี้ของภาษาซี ทำให้หมายความว่าแถวลำดับที่แท้จริงและแถวลำดับที่จัดสรรอย่างพลวัตเสมือนใช้แทนกันได้ เนื่องด้วยแถวลำดับเข้าถึงผ่านตัวชี้เสมอ (ในทางปฏิบัติ) การเข้าถึงแถวลำดับจึงไม่มีการตรวจสอบขนาดภายในแถวลำดับ แม้ว่าตัวแปลโปรแกรมอาจมีตัวเลือกสำหรับตรวจสอบขอบเขตก็ตาม การใช้งานเกินขอบเขตของแถวลำดับจึงยังคงสามารถเป็นไปได้ ซึ่งเกิดขึ้นค่อนข้างเป็นปกติในรหัสที่เขียนอย่างไม่ระมัดระวัง และนำไปสู่ผลสะท้อนกลับหลายอย่าง อาทิ การเข้าถึงหน่วยความจำที่ไม่อนุญาต การทำให้ข้อมูลผิดแปลกไป บัฟเฟอร์ส่วนเกิน และสิ่งผิดปกติขณะทำงาน

ถึงแม้ภาษาซีรองรับแถวลำดับแบบสถิต แต่ก็ไม่จำเป็นว่าดัชนีของแถวลำดับจะต้องมีผล (การตรวจสอบขอบเขต) ตัวอย่างเช่น เราสามารถลองบันทึกค่าสมาชิกตัวที่หกในแถวลำดับที่มีสมาชิกห้าตัวได้ ซึ่งจะทำให้เกิดผลที่ไม่คาดคิด ความผิดพลาดเช่นนี้เรียกว่า *บัฟเฟอร์ส่วนเกิน* (buffer overflow/overrun) เป็นสาเหตุที่สำคัญอย่างหนึ่งของปัญหาด้านความปลอดภัย เนื่องจากเทคโนโลยีการจัดการตรวจสอบขอบเขต (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 หรือไม่เช่นนั้นก็ต้องผ่านค่าไปพร้อมกับข้อมูลอื่นที่จำเป็นต้องทราบ นอกจากนี้ แถวลำดับของแถวลำดับที่จัดสรรขนาดแบบพลวัต ไม่สามารถเข้าถึงได้โดยใช้ดัชนีสองชั้น (ตัวอย่างกรณีนี้เช่นการจัดสรรแถวลำดับด้วย "เวกเตอร์แถว" ของตัวชี้ไปยังสตริง)

ภาษาซี99 ได้แนะนำ "แถวลำดับความยาวแปรได้" เพิ่มเข้ามา แต่ก็ยังมีปัญหาบางประการที่เหมือนกับปัญหาแถวลำดับของภาษาซี

คุณลักษณะเด่นชัดของภาษาซี (ซึ่งอาจทำให้สับสนด้วย) คือการปฏิบัติต่อแถวลำดับและตัวชี้ สัญกรณ์แถวลำดับ `x[i]` สามารถใช้กับตัวชี้ `x` ได้ โดยแปลความหมายว่าเป็นการเข้าถึงวัตถุตัว

ที่ $i + 1$ ของวัตถุข้อมูลที่อยู่ติดกันถัดจากตำแหน่งที่ x ซึ่งถือว่าเป็นสมาชิกตัวแรกของแถวลำดับ ($x[0]$)

$x[i]$ มีความหมายเทียบเท่า $*(x + i)$ ตามรูปแบบ และเนื่องจากชนิดตัวแปรของตัวชี้เป็นที่ยืดหยุ่นได้ ตำแหน่ง $x + i$ ที่ชี้ไปมีได้หมายความว่าจากตำแหน่ง x แล้วเพิ่มไปอีก i ไบต์ แต่หมายถึงเพิ่มไปอีก i คูณด้วยขนาดของสมาชิกที่ตำแหน่ง x ขนาดของสมาชิกนี้ได้มาจากการใช้ตัวดำเนินการ `sizeof` บนสมาชิกที่อ้างอิงกลับตัวใดตัวหนึ่งของ x ดังเช่น `n = sizeof *x` หรือ `n = sizeof x[0]`

นอกจากนี้ในบริบทส่วนใหญ่ของนิพจน์ ชื่อของแถวลำดับจะถูกแปลงเป็นตัวชี้ที่ชี้ไปยังสมาชิกตัวแรกของแถวลำดับนั้น สิ่งนี้บอกเป็นนัยว่าแถวลำดับจะไม่ถูกคัดลอกข้อมูลไปทั้งหมด เมื่อนำไปตั้งชื่ออาร์กิวเมนต์ของฟังก์ชัน แต่จะมีเพียงแค่ตำแหน่งของสมาชิกตัวแรกเท่านั้นที่ส่งผ่านไป ดังนั้นถึงแม้ว่าการเรียกใช้ฟังก์ชันในภาษาซีจะตีความว่าส่งโดยให้ค่า (pass-by-value) แต่แถวลำดับนั้นส่งโดยอ้างอิง (pass-by-reference) ในทางปฏิบัติ

จำนวนสมาชิกของแถวลำดับ x ที่ได้ประกาศไว้แล้ว สามารถคำนวณได้จาก `sizeof x / sizeof x[0]`

การสาธิตอย่างหนึ่งที่น่าสนใจต่อความใช้แทนกันได้ระหว่างตัวชี้และแถวลำดับแสดงไว้ด้านล่าง การกำหนดค่าทั้งสี่มีความหมายเทียบเท่ากันและเป็นรหัสที่ใช้งานได้ภาษาซี

แม้ว่าการกำหนดค่าทั้งสี่เทียบเท่ากัน แต่มีเพียงแบบแรกเท่านั้นที่แสดงรูปแบบการลงรหัสที่ดี กรณีอื่นอาจพบได้ในรหัสภาษาซีที่ยุ่งเหยิง

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

ฟังก์ชันการทำงานหนึ่งที่สำคัญที่สุดของภาษาโปรแกรมคือการให้บริการการจัดการหน่วยความจำและวัตถุที่บันทึกอยู่ในหน่วยความจำ ภาษาซีมีสามแนวทางที่ต่างกันเพื่อจัดสรรหน่วยความจำสำหรับวัตถุ

- การจัดสรรหน่วยความจำสถิต ที่ว่างสำหรับวัตถุในรหัสฐานสองซึ่งเว้นไว้ขณะแปลโปรแกรม วัตถุเหล่านี้มีอายุขัย (extent) ครอบคลุมทั้งรหัสฐานสองที่มีวัตถุนั้นบรรจุลงในหน่วยความจำ

- การจัดสรรหน่วยความจำอัตโนมัติ วัตถุชั่วคราวสามารถเก็บบันทึกในกองซ้อน (stack) และที่ว่างนี้จะถูกเรียกคืนและใช้ใหม่หลังจากวัตถุที่ประกาศเลิกการทำงานโดยอัตโนมัติ
- การจัดสรรหน่วยความจำพลวัต บล็อกต่าง ๆ ของหน่วยความจำในขนาดที่ต้องการสามารถร้องขอได้ขณะทำงาน โดยใช้ฟังก์ชันไลบรารีอาทิ `malloc` จองเนื้อที่หน่วยความจำที่เรียกว่า ฮีป (heap) บล็อกเหล่านี้กองอยู่จนกว่าจะถูกเรียกคืนเพื่อใช้ใหม่โดยใช้ฟังก์ชัน `free` ในภายหลัง

แนวทางสามอย่างนี้เหมาะสมในสถานการณ์และข้อแลกเปลี่ยนที่ต่างกันไป ตัวอย่างเช่น การจัดสรรหน่วยความจำสถิตไม่มีการดำเนินงานสิ้นเปลือง (overhead) เพื่อการจัดสรร การจัดสรรหน่วยความจำอัตโนมัติอาจมีการสิ้นเปลืองน้อย และการจัดสรรหน่วยความจำพลวัตอาจเป็นไปได้ว่ามีความสิ้นเปลืองอย่างมากทั้งการจัดสรรและการเรียกคืน ในทางตรงข้าม ที่ว่างในกองซ้อนโดยทั่วไปมีขนาดจำกัดและไม่คงทนถาวรไปกว่าหน่วยความจำแบบสถิตหรือที่ว่างในฮีป และการจัดสรรหน่วยความจำพลวัตสามารถจัดสรรวัตถุที่ทราบขนาดเฉพาะขณะทำงานได้ โปรแกรมภาษาซีส่วนใหญ่จึงใช้งานทั้งสามแนวทางอย่างกว้างขวาง

การจัดสรรมักให้ความสำคัญแก่แบบอัตโนมัติหรือแบบสถิตมากกว่า เพราะตัวแปลโปรแกรมเป็นส่วนจัดการหน่วยเก็บบันทึก ทำให้โปรแกรมเมอร์ไม่ต้องจัดสรรและเรียกคืนหน่วยเก็บบันทึกจุกจิกด้วยตนเองซึ่งเสี่ยงต่อการเกิดข้อผิดพลาด อย่างไรก็ตาม โครงสร้างข้อมูลหลายชนิดสามารถขยายขนาดได้ในขณะทำงาน และเนื่องจากการจัดสรรสถิต (และการจัดสรรอัตโนมัติในภาษาซี89 และซี90) จะต้องมีขนาดตายตัวขณะแปลโปรแกรม หลายสถานการณ์จึงจำเป็นต้องใช้การจัดสรรพลวัต ก่อนที่จะมีมาตรฐานซี99 แถวลำดับความยาวแปรได้เป็นตัวอย่างปัญหาหนึ่งของกรณีนี้

วัตถุที่จัดสรรแบบอัตโนมัติและพลวัตจะถูกกำหนดค่าเริ่มต้นถ้าได้ระบุไว้ หรือมิฉะนั้นมันจะมีค่าที่ไม่แน่นอน (ไม่ว่ารูปแบบรหัสฐานสองบนหน่วยเก็บบันทึกจะเป็นอะไรก็ตาม ซึ่งอาจไม่เป็นค่าที่ใช้งานได้สำหรับชนิดตัวแปรนั้น) ถ้าโปรแกรมพยายามเข้าถึงค่าที่ไม่กำหนดนี้ ผลลัพธ์จะไม่สามารถนิยามได้ ตัวแปลโปรแกรมสมัยใหม่หลายโปรแกรมพยายามตรวจสอบและแจ้งเตือนเกี่ยวกับปัญหานี้ แต่ก็เกิดทั้งผลบวกและผลลบ

ปัญหาอีกประการหนึ่งคือ การจัดสรรหน่วยความจำในฮีปจำเป็นต้องกระทำพร้อมกับการใช้งานจริงด้วยตนเองในโปรแกรมใด ๆ ก็ตาม เพื่อให้มันสามารถนำกลับมาใช้ใหม่มากที่สุดเท่าที่เป็นไปได้ ตัวอย่างเช่น ถ้ามีตัวชี้ไปยังฮีปที่ถูกจัดสรรนอกขอบเขต หรือค่าของตัวชี้ถูกเขียนทับก่อนเรียกใช้ `free` จะทำให้หน่วยความจำที่ตำแหน่งนั้นไม่สามารถเรียกคืนเพื่อใช้ใหม่ภายหลังและสูญเสียไปกับโปรแกรม อันเป็นปรากฏการณ์ที่เรียกว่า หน่วยความจำรั่ว (memory leak) ในทางกลับกัน การปลดปล่อยหน่วยความจำเร็วเกินไปแล้วยังคงใช้งานอยู่ซึ่งเป็นไปได้ แต่เนื่องจากระบบ

จัดสรรหน่วยความจำสามารถจัดสรรอีกครั้งหรือใช้หน่วยความจำที่ถูกทำให้ว่าง พฤติกรรมที่คาดเดาไม่ได้ก็อาจเกิดขึ้น โดยปกติอาการจะปรากฏในส่วนของโปรแกรมที่อยู่ไกลจากจุดที่ทำให้เกิดความผิดพลาดจริง ทำให้ตรวจแก้ปัญหาลำบาก ปัญหาเช่นนี้ได้รับการปรับปรุงแก้ไขในภาษาโปรแกรมที่มีการเก็บกวาดข้อมูลขยะอัตโนมัติ

ภาษาซีใช้ไลบรารีเป็นวิธีการหลักสำหรับส่วนขยาย ไลบรารีคือกลุ่มของฟังก์ชันที่บรรจุอยู่ในไฟล์เดียวกันโดย "ถาวร" ไลบรารีแต่ละชนิดจะมีไฟล์ส่วนหัว ซึ่งรวบรวมต้นแบบ (prototype) ตามฟังก์ชันที่มีอยู่ในไลบรารีซึ่งอาจถูกเรียกใช้โดยโปรแกรม และมีการประกาศชนิดข้อมูลพิเศษและสัญลักษณ์แมโครที่ใช้ในฟังก์ชันเหล่านั้น โปรแกรมจะต้องรวมไฟล์ส่วนหัวนี้เข้าไปเพื่อใช้งานไลบรารี และไลบรารีจะต้องเชื่อมโยงกับโปรแกรม ซึ่งในหลายกรณีอาจต้องใช้ตัวบ่งชี้คอมไพเลอร์ (compiler flag) (เช่น `-lm` สำหรับไลบรารีคณิตศาสตร์เป็นต้น)

ไลบรารีสามัญที่สุดคือไลบรารีมาตรฐานของภาษาซี ซึ่งระบุไว้โดยมาตรฐานไอโซและแอนซีซีและติดมากับทุกโปรแกรมที่พัฒนาด้วยภาษาซี (ส่วนการพัฒนาบนสภาพแวดล้อมแบบฝังตัวอาจมีไลบรารีมาตรฐานเพียงส่วนย่อยส่วนหนึ่ง) ไลบรารีนี้รองรับกระแสข้อมูลรับเข้าและส่งออก การจัดสรรหน่วยความจำ คณิตศาสตร์ สายอักขระ และค่าของเวลา

ไลบรารีสามัญอีกกลุ่มหนึ่งเป็นฟังก์ชันที่เจาะจงใช้กับโปรแกรมที่ทำงานบนระบบปฏิบัติการยูนิกซ์หรือคล้ายยูนิกซ์ โดยเฉพาะฟังก์ชันที่มีส่วนต่อประสานเข้ากับเคอร์เนล ฟังก์ชันเหล่านี้ได้ให้รายละเอียดไว้ในมาตรฐานหลากหลายเช่น โพสซิกซ์หรือข้อกำหนดคุณลักษณะยูนิกซ์เชิงเดี่ยว (Single UNIX Specification)

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

เครื่องมือหลายอย่างถูกสร้างขึ้นเพื่อช่วยเหลือโปรแกรมเมอร์ภาษาซี เพื่อหลีกเลี่ยงปัญหาบางประเภทที่มากับภาษา เช่น ข้อความสั่งที่มีพฤติกรรมไม่นิยาม หรือข้อความสั่งที่ปฏิบัติไม่ดีซึ่งอาจส่งผลให้เกิดพฤติกรรมที่ไม่ตั้งใจหรือความผิดพลาดขณะทำงาน

การตรวจสอบแก้ไขรหัสต้นฉบับอัตโนมัติเป็นประโยชน์สำหรับทุกภาษา และภาษาซีก็มีเครื่องมือเช่นกันเช่น lint การใช้ lint โดยปกติเพื่อตรวจจบบั๊กที่น่าสงสัยเมื่อโปรแกรมเขียนขึ้นเป็นครั้งแรก เมื่อโปรแกรมผ่านการตรวจสอบจาก lint แล้ว มันจึงจะถูกแปลด้วยตัวแปลภาษาซี ตัวแปลภาษาหลายตัวก็สามารถเลือกได้เพื่อแจ้งเตือน เกี่ยวกับโครงสร้างที่ถูกต้องตามวากยสัมพันธ์แต่อาจเกิดความผิดพลาดได้จริง มีสราซี เป็นกลุ่มแนวทางที่มีกรรมสิทธิ์เพื่อการหลีกเลี่ยงรหัสที่น่าสงสัยเช่นนั้น ซึ่งพัฒนาขึ้นสำหรับระบบฝังตัว

นอกจากนี้ยังมีตัวแปลโปรแกรม ไบรารี และกลไกระดับระบบปฏิบัติการ เพื่อการตรวจสอบขอบเขตของแถวลำดับ การตรวจจับบัฟเฟอร์ส่วนสั้น การทำให้เป็นอนุกรม (serialization) และการเก็บกวาดข้อมูลขยะอัตโนมัติ ซึ่งมีใช้ส่วนหนึ่งที่เป็นมาตรฐานของภาษาซี

เครื่องมืออื่นอย่างเช่น เพียวริฟาย แวลกรินด์ และการเชื่อมโยงกับไลบรารีที่มีฟังก์ชันจัดสรรหน่วยความจำแบบพิเศษ สามารถช่วยเปิดเผยข้อผิดพลาดในหน่วยความจำขณะทำงานได้

ภาษาซีมีอิทธิพลต่อภาษาอื่นในยุคหลังทั้งในทางตรงและทางอ้อมเช่น ภาษาจาวา ภาษาเพิร์ล ภาษาพีเอชพี จาวาสคริปต์ ภาษาแอลพีซี ภาษาซีชาร์ป และซีเชลล์ของยูนิกซ์ อิทธิพลที่แพร่หลายมากที่สุดคือรูปแบบวากยสัมพันธ์ ทุกภาษาที่กล่าวมาได้รวมวากยสัมพันธ์ของข้อความสั่งกับนิพจน์ของภาษาซี พร้อมทั้งระบบชนิดตัวแปร อันเป็นตัวแบบข้อมูลและ/หรือโครงสร้างโปรแกรมขนาดใหญ่ที่ต่างไปจากของภาษาซี ซึ่งบางครั้งก็ต่างกันอย่างมาก

เมื่อแนวคิดภาษาเชิงวัตถุเป็นที่นิยม ภาษาซีพลัสพลัสและภาษาอ็อบเจกทีฟ-ซีเป็นส่วนขยายที่แตกต่างกันของภาษาซีที่ให้ความสามารถเชิงวัตถุได้ ภาษาทั้งสองแต่เดิมทำให้เกิดผลโดยใช้ตัวแปลภาษาแบบแปลรหัสต่อรหัส นั่นคือรหัสต้นฉบับของภาษาดังกล่าวจะถูกแปลเป็นรหัสภาษาซีก่อน จากนั้นจึงแปลด้วยคอมไพเลอร์อีกต่อหนึ่ง

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

ภาษาอ็อบเจกทีฟ-ซีเดิมเป็นเพียง "ชั้นบาง ๆ" บนภาษาซีและยังคงครอบคลุมภาษาซีอย่างเข้มงวด ซึ่งอนุญาตให้เขียนโปรแกรมเชิงวัตถุโดยใช้กระบวนการทศนิยมชนิดตัวแปรผสมพลวัต/สถิตวากยสัมพันธ์ของภาษาอ็อบเจกทีฟ-ซีมาจากทั้งภาษาซีและภาษาสมอลล์ทอล์ก นั่นคือวากยสัมพันธ์ที่เกี่ยวข้องกับการประมวลผลก่อน นิพจน์ การประกาศฟังก์ชัน และการเรียกใช้ฟังก์ชันรับมาจากภาษาซี ในขณะที่วากยสัมพันธ์สำหรับคุณลักษณะเชิงวัตถุนำมาจากภาษาสมอลล์ทอล์ก

ภาษาดีทำคุณลักษณะหลายอย่างให้ต่างออกไปแต่ยังคงไว้ซึ่งวากยสัมพันธ์ทั่วไปของภาษาซี ไม่เหมือนภาษาซีพลัสพลัสที่แทบจะเข้ากันได้แบบย้อนหลังกับภาษาซี ภาษาดีละทิ้งคุณลักษณะจำนวนหนึ่งของภาษาซีออกไป เนื่องจากวอลเตอร์ ไบรต์ (Walter Bright) ผู้ออกแบบภาษาดีพิจารณาว่าไม่มีความจำเป็นต้องใช้คุณลักษณะเหล่านั้น รวมทั้งตัวประมวลผลก่อนและไดรอักษร ส่วนขยายบางอย่างของภาษาดีไปยังภาษาซี ทับซ้อนกับส่วนขยายไปยังภาษาซีพลัสพลัส


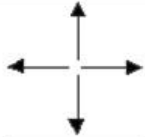






ภาษาลิมโบเป็นภาษาหนึ่งที่พัฒนาโดยทีมงานที่เบลล์แล็บส์ และในขณะที่ยังคงรักษา วากยสัมพันธ์และลักษณะทั่วไปบางอย่างของภาษาซี ก็ยังมีการเก็บกวาดข้อมูลขยะและภาวะพร้อม กันที่มีพื้นฐานบนกระบวนการสื่อสารแบบลำดับ (communicating sequential processes)

ภาษาไพทอนสืบทอดมาจากภาษาซีในแนวทางที่ต่างออกไป ในขณะที่วากยสัมพันธ์และความหมายของภาษาไพทอนแตกต่างกับภาษาซีอย่างสิ้นเชิง แต่เครื่องมือทำให้เกิดผลในภาษาไพทอนที่ใช้กันอย่างกว้างขวางที่สุดคือซีไพทอน ซึ่งเป็นโปรแกรมภาษาซีแบบโอเพนซอร์ซ สิ่งนี้ช่วยให้ผู้ใช้สามารถเขียนภาษาซีเป็นส่วนขยายของภาษาไพทอน หรือฝังภาษาไพทอนลงในโปรแกรมภาษาซี ความสัมพันธ์อย่างใกล้ชิดนี้เป็นปัจจัยหนึ่งที่น่าไปสู่ความสำเร็จของภาษาไพทอนในฐานะภาษาพลวัตเพื่อการใช้งานทั่วไป

ภาษาเพิร์ลเป็นอีกตัวอย่างหนึ่งของภาษาโปรแกรมที่มีต้นกำเนิดจากภาษาซี โครงสร้างโดยรวมทั้งหมดของภาษาเพิร์ลมาจากภาษาซีอย่างมาก เครื่องมือทำให้เกิดผลของภาษาเพิร์ลมาตรฐานเขียนขึ้นด้วยภาษาซี และรองรับส่วนขยายที่เขียนในภาษาซีด้วย

2.4 ผังงาน (Flowchart)

คือ แผนภาพแสดงลำดับขั้นตอนการทำงานของ Algorithm, Workflow, Process เป็นเครื่องมือใช้การรวบรวมจัดลำดับความคิด เพื่อให้เห็นขั้นตอนการทำงานที่ชัดเจนและใช้วางแผนการทำงานขั้นแรก โดยสัญลักษณ์ Flowchart แสดงถึงการทำงานลักษณะต่างๆ เชื่อมต่อกัน Flowchart ถูกใช้ในการออกแบบ เพื่อช่วยให้เห็นภาพสิ่งที่เกิดขึ้นและช่วยให้เข้าใจ

สัญลักษณ์	ความหมายและการใช้
	จุดเริ่มต้น / สิ้นสุดของโปรแกรม
	ลูกศรแสดงทิศทางการทำงานของโปรแกรมและการไหลของข้อมูล
	ใช้แสดงคำสั่งในการประมวลผล หรือการกำหนดค่าข้อมูลให้กับตัวแปร
	แสดงการอ่านข้อมูลจากหน่วยเก็บข้อมูลสำรองเข้าสู่หน่วยความจำหลัก ภายในเครื่องหรือการแสดงผลลัพธ์จากการประมวลผลออกมา
	การตรวจสอบเงื่อนไขเพื่อตัดสินใจ โดยจะมีเส้นออกจากรูปเพื่อแสดงทิศทางการทำงานต่อไป เงื่อนไขเป็นจริงหรือเป็นเท็จ
	แสดงผลหรือรายงานที่ถูกสร้างออกมา
	แสดงจุดเชื่อมต่อของผังงานภายใน หรือเป็นที่บรรจบของเส้นหลายเส้น ที่มาจากหลายทิศทางเพื่อจะไปสู่การทำงานอย่างใดอย่างหนึ่งที่เหมือนกัน
	การขึ้นหน้าใหม่ ในกรณีที่ผังงานมีความยาวเกินกว่าที่จะแสดงพอในหนึ่งหน้า

รูปที่ 2.5 สัญลักษณ์ ผังงาน (Flowchart) (<http://www.thaiall.com/flowchart/indexo.html>)

2.4.1 ประโยชน์ของผังงาน

- 1) ช่วยอธิบายลำดับขั้นตอนการทำงานของโปรแกรม
- 2) ทำให้ตรวจสอบข้อผิดพลาดของโปรแกรมได้ง่าย
- 3) ทำให้ผู้อื่นสามารถศึกษาการทำงานของโปรแกรมและแก้ไขโปรแกรมได้ง่าย

ในการเขียนโปรแกรมด้วยภาษาคอมพิวเตอร์ โดยทั่วไปจะมีหลักการหรือขั้นตอนที่สำคัญทั้งหมด 5 ขั้นตอนได้แก่

- 1) การวิเคราะห์ปัญหา
- 2) การออกแบบโปรแกรม
- 3) การเขียนโปรแกรม
- 4) การทดสอบและแก้ไขโปรแกรม
- 5) ทำเอกสารประกอบโปรแกรม

ดังนั้นก่อนที่จะลงมือเขียนโปรแกรมในขั้นตอนที่ 3 หลังจากทำการวิเคราะห์ปัญหาแล้ว จะต้องมีการออกแบบโปรแกรมเพื่อเป็นการวางแผนการทำงานก่อน ผังงาน Flowchart เป็นเครื่องมือหนึ่งที่ใช้อธิบายลำดับขั้นตอนการทำงานในรูปแบบแผนภาพ โดยใช้สัญลักษณ์รูปร่างต่าง ๆ ที่มีความหมายแทนคำสั่ง และใช้ข้อความในสัญลักษณ์แทนข้อมูลตัวแปร ตัวดำเนินการทางการคำนวณ และการเปรียบเทียบ นอกจากนั้นผังงานยังใช้แสดงความสัมพันธ์ระหว่างขั้นตอนการทำงานต่าง ๆ โดยสามารถแบ่งลักษณะการทำงานและความสัมพันธ์เป็นรูปแบบต่างๆ ได้แก่ การทำงานแบบมีลำดับ การทำงานแบบมีเงื่อนไข และการทำงานแบบทวนซ้ำภายใต้เงื่อนไขต่าง ๆ หลังจากนั้นจึงนำผังงาน Flowchart ที่ออกแบบไว้มาไปเขียนเป็นภาษาคอมพิวเตอร์ ดังนั้นผังงานจึงเป็นเครื่องมือที่ช่วยให้ผู้เขียนโปรแกรมและผู้ใช้ สามารถมองเห็นภาพการทำงานของโปรแกรมที่กำลังจะสร้างได้อย่างเป็นระบบและง่ายขึ้น

2.4.2 ประเภทของผังงาน โดยทั่วไปผังงานคอมพิวเตอร์แบ่งเป็น 2 ประเภทใหญ่

1) ผังงานระบบ (System Flowchart)

เป็นผังงานที่แสดงถึงขั้นตอนการทำงานภายในระบบหนึ่ง ๆ เพื่อให้เห็นโครงสร้างโดยภาพรวมของระบบ ซึ่งจะแสดงถึงความเกี่ยวข้องของส่วนที่สำคัญต่างๆ ในระบบนั้น เช่น เอกสารข้อมูลเบื้องต้น สื่อบันทึกข้อมูลที่ใช้ ข้อมูลจะส่งผ่านไปยังหน่วยงานใด มีกิจกรรมประมวลผลข้อมูลอะไรในหน่วยงานนั้น แล้วจะส่งต่อไปหน่วยงานใด เป็นต้น ดังนั้นผังงานระบบอาจเกี่ยวข้อง กับข้อมูล สื่อหรือแหล่งบันทึกข้อมูล วัสดุอุปกรณ์ คน หรือฝ่ายงานที่เกี่ยวข้อง ซึ่งแต่ละจุดจะประกอบไปด้วย

การนำเข้าข้อมูล วิธีการประมวลผล และการแสดงผลลัพธ์ (Input – Process - Output)

2) ผังงานโปรแกรม (Program Flowchart) หรือเรียกสั้นๆ ว่า ผังงาน

ผังงานประเภทนี้แสดงถึงขั้นตอนของคำสั่งที่ใช้ในโปรแกรม ผังงานนี้อาจสร้างจากผังงานระบบโดยผู้เขียนผังงานจะดึงเอาแต่ละจุด ที่เกี่ยวข้องกับการทำงานของเครื่องคอมพิวเตอร์ที่ปรากฏในผังงานระบบมาเขียน เพื่อให้ทราบว่าถ้าจะใช้คอมพิวเตอร์ทำงานควรที่จะมีขั้นตอนคำสั่งอย่างไร เพื่อให้ได้ผลลัพธ์ตามที่ต้องการ และจะได้นำมาเขียนโปรแกรมคอมพิวเตอร์ต่อไป

ดังนั้นการเขียนผังงานก็จะมีประโยชน์ เหมาะสำหรับผู้บริหาร ผู้วิเคราะห์ระบบ ผู้เขียนโปรแกรม และบุคคลอื่นที่ต้องการศึกษา ทาให้ทราบถึงความสัมพันธ์ของระบบตั้งแต่เริ่มต้น ว่ามีการปฏิบัติแต่ละขั้นตอนอย่างไร ใช้วิธีการอะไรบ้าง สุดท้ายจะได้ผลลัพธ์อะไรบ้าง เมื่อเข้าใจระบบงานหรือสิ่งที่กำลังศึกษาก็จะช่วยให้สามารถปฏิบัติงานและแก้ปัญหาได้อย่างมีประสิทธิภาพมากยิ่งขึ้น

ประโยชน์ของผังงาน

ผังงานเป็นเครื่องมือที่ช่วยให้การศึกษาลาดับขั้นตอนของโปรแกรมน่าง่ายขึ้น จึงนิยมเขียนผังงานประกอบการเขียนโปรแกรม ด้วยเหตุผลดังนี้

1) คนส่วนใหญ่สามารถเรียนรู้และเข้าใจผังงานได้ง่าย เพราะผังงานไม่ขึ้นอยู่กับภาษาคอมพิวเตอร์ภาษาใดภาษาหนึ่ง เป็นเครื่องมือที่ใช้ในการสื่อสารได้ทุกภาษา

2) ผังงานเป็นการสื่อความหมายด้วยภาพ ช่วยลำดับขั้นตอนการทำงานของโปรแกรมให้ง่ายและสะดวกต่อการทำความเข้าใจ สามารถนำไปเขียนโปรแกรมได้โดยไม่สับสน ซึ่งถ้าหากใช้ข้อความหรือคำพูดอาจจะสื่อความหมายผิดไปได้

3) ในงานโปรแกรมที่ไม่สลับซับซ้อน ช่วยในการตรวจสอบความถูกต้องของลำดับขั้นตอน และแก้ไขโปรแกรมได้ง่าย เมื่อเกิดข้อผิดพลาด

4) ช่วยให้ผู้อื่นสามารถศึกษาการทำงานของโปรแกรมได้อย่างง่าย สะดวก และรวดเร็วมากขึ้น

5) การบำรุงรักษาโปรแกรมหรือการเปลี่ยนแปลงแก้ไขโปรแกรมในภายหลัง ให้มีประสิทธิภาพ ถ้าพิจารณาจากผังงานจะช่วยให้สามารถทบทวนงานในโปรแกรมก่อนปรับปรุงแก้ไขได้สะดวกและง่ายขึ้น

ข้อจำกัดของการเขียนผังงาน

นักเขียนโปรแกรมบางคนไม่นิยมการเขียนผังงานก่อนที่จะเขียนโปรแกรม เพราะเสียเวลาในการเขียนเป็นรูปภาพหรือสัญลักษณ์ต่างๆ นอกจากนี้ยังมีเหตุผลอื่นๆ ได้แก่

1) ผังงานเป็นการสื่อความหมายระหว่างบุคคลต่อบุคคลมากกว่าที่จะสื่อความหมายบุคคลกับเครื่องคอมพิวเตอร์ เพราะผังงานไม่ขึ้นอยู่กับภาษาคอมพิวเตอร์ ภาษาใดภาษาหนึ่ง ทาให้เครื่องคอมพิวเตอร์ไม่สามารถรับรู้และเข้าใจว่าผังงานต้องการอะไร

2) ผังงานไม่สามารถแทนลักษณะคำสั่งของภาษาคอมพิวเตอร์บางคำสั่งได้อย่างชัดเจน

3) กรณีที่งานมีขนาดใหญ่ ผังงานจะมีขนาดใหญ่ด้วย ถ้ามีการเปลี่ยนแปลงแก้ไขจะทำได้ยาก ควรเขียนแยกเป็นส่วน ๆ แล้วค่อยสร้างจุดเชื่อมโยงในแต่ละส่วน

4) การเขียนผังงานอาจเป็นการเปลี่ยนแปลงกระดาษและอุปกรณ์อื่นๆ ประกอบการเขียนภาพ ทั้ง ๆ ที่การอธิบายงานหรือการเขียนโปรแกรมจะใช้เนื้อที่เพียง 3 - 4 บรรทัดเท่านั้น
วิธีการเขียนผังงานที่ดี

การเขียนผังงานควรคำนึงถึงสิ่งต่าง ๆ ดังนี้

- 1) ใช้สัญลักษณ์ตามที่กำหนดไว้
- 2) ผังงานจะต้องมีจุดเริ่มต้น (Start) และสิ้นสุด (Stop/End/Finish)
- 3) ใช้หัวลูกศรแสดงทิศทางการไหลของข้อมูลจากบนลงล่างหรือซ้ายไปขวา (ยกเว้นที่ต่อทางซ้าย)
- 4) ทุกแผนภาพต้องมีลูกศรแสดงทิศทางเข้า 1 เส้นและออก 1 เส้นโดยไม่มีการปล่อยจุดใดจุดหนึ่งไว้
- 5) เขียนคำอธิบายการทำงานในแต่ละขั้นตอนโดยใช้ข้อความที่สั้น กระชับรัด ชัดเจนและเข้าใจได้ง่าย
- 6) ควรหลีกเลี่ยงโยงเส้นไปมาทำให้เกิดจุดตัดมากเพราะจะทำให้เกิดข้อผิดพลาดง่าย ควรใช้สัญลักษณ์เชื่อมจุดต่อเนื่องแทน
- 7) ไม่ควรโยงเส้นเชื่อมผังงานที่อยู่ไกลมาก ๆ ควรใช้สัญลักษณ์จุดเชื่อมต่อแทน
- 8) ผังงานที่ดีควรมีความเป็นระเบียบเรียบร้อย สะอาดและชัดเจน สามารถเข้าใจและติดตามขั้นตอนได้ง่าย
- 9) ผังงานควรมีการทดสอบความถูกต้องของการทำงานก่อนไปเขียนโปรแกรมหลักในการเขียนโปรแกรม

ในการเขียนผังงานหรือโปรแกรม ต้องรู้จักเลือกใช้รูปภาพหรือสัญลักษณ์ที่เหมาะสม รวมถึงอุปกรณ์ที่ช่วยในการเขียนผังงานที่เรียกว่า “ Flow Chart Template “ ซึ่งอุปกรณ์นี้จะช่วยให้การเขียนผังงานสะดวกและรวดเร็วยิ่งขึ้น

ในการเขียนผังงานนี้จะเขียนตามขั้นตอนและวิธีการประมวลผลที่ได้ทำการวิเคราะห์งานเอาไว้แล้ว ซึ่งต้องพิจารณาตามลำดับก่อนหลังของการทำงาน เพื่อจัดภาพของผังงานให้เป็นมาตรฐานง่ายต่อการเข้าใจ และช่วยให้การเขียนโปรแกรมจากผังงานมีประสิทธิภาพมากยิ่งขึ้น เพื่อให้การเขียนผังงานเป็นมาตรฐานเดียวกันจะใช้ลำดับในการเขียนผังงานดังนี้

1) การกำหนดค่าเริ่มต้น เป็นการกำหนดค่าเริ่มต้นให้กับตัวแปรต่างๆ ที่จำเป็นบางตัว ได้แก่ ตัวแปรที่ใช้เป็นตัวนับ หรือตัวแปรที่เป็นตัวคำนวณผลรวมต่างๆ

2) การรับข้อมูลเข้า เป็นการรับข้อมูลนำเข้ามาจากแหล่งข้อมูลต่างๆ เพื่อใช้เป็นส่วนหนึ่งของการประมวลผล แล้วนำค่ามาเก็บไว้ในตัวแปรใด ๆ ที่กำหนดเอาไว้

3) การประมวลผล เป็นการประมวลผลตามที่ได้มีการกำหนด หรือเป็นการคำนวณต่างๆ ซึ่งจะต้องทำทีละลำดับขั้นตอนและแยกรูปแต่ละรูปออกจากกันให้ชัดเจนด้วย

4) การแสดงผลลัพธ์ เป็นการแสดงข้อมูลที่ได้จากการคำนวณหรือผลลัพธ์ที่ต้องการหรือค่าจากตัวแปรต่างๆ ซึ่งการแสดงผลลัพธ์นี้มักจะกระทำหลังจากการประมวลผลหรือหลังจากการรับข้อมูลเข้ามาแล้ว

ข้อสังเกตในการเขียนโฟลว์ชาร์ต

- 1) โฟลว์ชาร์ตใด ๆ จะมีจุดเริ่มต้นและจุดสิ้นสุดเพียงอย่างละแห่งเดียวเท่านั้น
- 2) ทุกสัญลักษณ์ที่ใช้แทนขั้นตอนการทำงาน จะต้องมียศทางเข้าเพียง 1 แห่งและทิศทางออกเพียง 1 แห่งเท่านั้น ยกเว้นสัญลักษณ์ของจุดเริ่มเริ่มต้น จุดสิ้นสุด จุดต่อ และ การตัดสินใจ
- 3) ทิศทางของลำดับขั้นตอนการทำงานในโฟลว์ชาร์ตนิยมเขียนจากซ้ายไปขวา หรือจากบนลงล่าง
- 4) หลีกเลี่ยงการขีดเส้นโยงไปโยงมาในลักษณะที่ตัดกัน ถ้าจำเป็นต้องโยงเส้นดังกล่าวถึงกันควรใช้เครื่องหมายต่อจุดเพื่อเชื่อมความสัมพันธ์แทน
- 5) สัญลักษณ์ต่าง ๆ นั้นจะเปลี่ยนรูปเป็นอย่างอื่นไม่ได้ ต้องเป็นไปตามมาตรฐานที่กำหนดไว้แล้วเท่านั้น
- 6) ควรมีเครื่องหมายลูกศรกำกับทิศทางการไหลให้กับแต่ละสัญลักษณ์ ด้วย
- 7) คำอธิบายการทำงานควรเขียนให้สั้นเข้าใจง่ายและเขียนในสัญลักษณ์ของโฟลว์ชาร์ต ทั้งหมดหากมีคำอธิบายเพิ่มเติมให้เขียนไว้บนสัญลักษณ์ด้านขวา
- 8) ในการเขียนโฟลว์ชาร์ตควรเขียนให้เป็นระเบียบ เรียบร้อย และสะอาด

2.5 โปรแกรม Arduino

Arduino เป็นโปรแกรมไมโครคอนโทรลเลอร์บอร์ดเดียวที่พัฒนาขึ้นครั้งแรกในปี 2548 และเริ่มมีชีวิตในฐานะโครงการนักเรียนในราคาเพียง Arduino ใช้เพื่อทำให้การใช้งานแอปพลิเคชันแบบโต้ตอบและโปรแกรมอื่น ๆ ง่ายขึ้นและเข้าถึงได้มากขึ้นสำหรับผู้ใช้อ Arduino ช่วยให้ผู้ใช้อัปโหลดโปรแกรมไปยังไมโครคอนโทรลเลอร์โดยไม่จำเป็นต้องใช้ชิปหรืออุปกรณ์อื่น Arduino มีชุดเครื่องมือและคุณสมบัติต่าง ๆ มากมายและมาพร้อมกับบอร์ดฮาร์ดแวร์โอเพนซอร์สซึ่งใช้ไมโครคอนโทรลเลอร์ Atmel AVR 8 บิต บอร์ดฮาร์ดแวร์ Arduino สามารถประกอบไว้ล่วงหน้าหรือสามารถประกอบเองได้โดยผู้ใช้ ตัวเลือกหลังเป็นตัวเลือกที่ถูกกว่าและเป็นฟรอนท์ที่เป็นที่นิยมสำหรับผู้ใช้ที่ชอบคนจรจัดที่มีอุปกรณ์อิเล็กทรอนิกส์และชอบที่จะสร้างสิ่งต่าง ๆ ด้วยตัวเอง โปรแกรม do-it-yourself มาพร้อมกับข้อมูลที่ครอบคลุมซึ่งสามารถแนะนำผู้ใช้ตลอดกระบวนการสร้างและช่วยเหลือพวกเขาหากพวกเขาคิดอยู่ที่จุดใด

2.6 การเขียนโปรแกรมคอมพิวเตอร์ (Computer programming)

เรียกให้สั้นลงว่า การเขียนโปรแกรม (Programming) หรือ การเขียนโค้ด (Coding) เป็นขั้นตอนการเขียน ทดสอบ และดูแลซอร์สโค้ดของโปรแกรมคอมพิวเตอร์ ซึ่งซอร์สโค้ดนั้นจะเขียนด้วยภาษาโปรแกรม ขั้นตอนการเขียนโปรแกรมต้องการความรู้ในหลายด้านด้วยกัน เกี่ยวกับโปรแกรมที่ต้องการจะเขียน และขั้นตอนวิธีที่จะใช้ ซึ่งในวิศวกรรมซอฟต์แวร์นั้น การเขียนโปรแกรมถือเป็นเพียงขั้นหนึ่งในวงจรชีวิตของการพัฒนาซอฟต์แวร์ การเขียนโปรแกรมจะได้มาซึ่งซอร์สโค้ดของโปรแกรมนั้นๆ โดยปกติแล้วจะอยู่ในรูปแบบของ ข้อความธรรมดา ซึ่งไม่สามารถนำไปใช้งานได้ จะต้องผ่านการคอมไพล์ตัวซอร์สโค้ดนั้นให้เป็นภาษาเครื่อง (Machine Language) เสียก่อนจึงจะได้เป็นโปรแกรมที่พร้อมใช้งาน

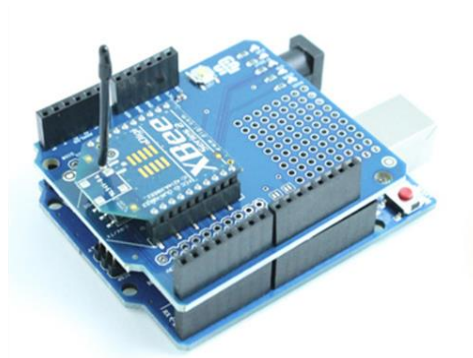
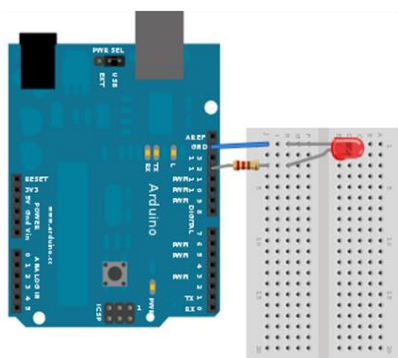
2.7 หลักการทำงานของบอร์ด Arduino

2.7.1 Arduino คืออะไร

Arduino (อาดูยโน) เป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลงเพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ด หรือโปรแกรมต่อได้อีกด้วย

ความง่ายของบอร์ด Arduino ในการต่ออุปกรณ์เสริมต่าง ๆ คือผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อมต่อเข้ากับขา I/O ของบอร์ด หรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม (Arduino Shield) ประเภทต่าง ๆ เช่น Arduino XBee Shield, Arduino Music

Shield, Arduino Relay Shield, Arduino Wireless Shield, Arduino GPRS Shield เป็นต้น มาเปรียบกับบอร์ดบนบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้เลย



รูปที่ 2.6 บอร์ด Arduino (http://fa.lnwfile.com/_/fa/_raw/3x/ei/ty.jpg)

2.7.2 จุดเด่นที่ทำให้บอร์ด Arduino เป็นที่นิยม

2.7.2.1 ง่ายต่อการพัฒนามีรูปแบบคำสั่งพื้นฐาน ไม่ซับซ้อนเหมาะสำหรับผู้เริ่มต้น

2.7.2.2 มี Arduino Community กลุ่มคนที่ร่วมกันพัฒนาที่แข็งแกร่ง

2.7.2.3 Open Hardware ทำให้ผู้ใช้สามารถนำบอร์ดไปต่อยอดใช้งานได้หลายด้าน

2.7.2.4 ราคาไม่แพง

2.7.2.5 Cross Platform สามารถพัฒนาโปรแกรมบน OS ใดก็ได้



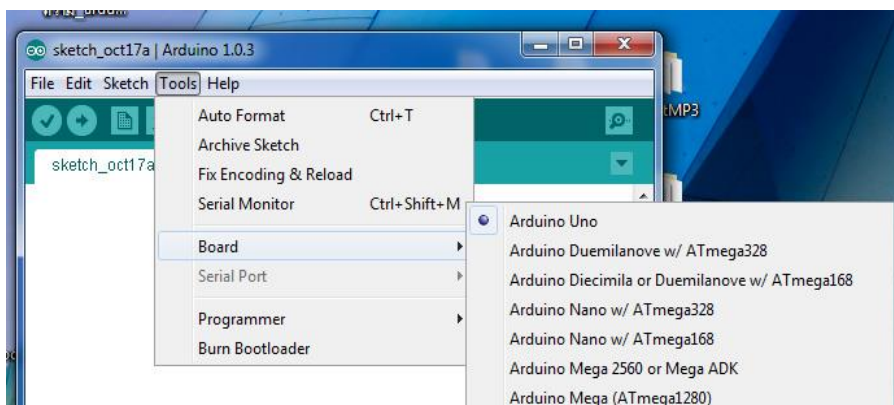
รูปที่ 2.7 รูปการเขียนโปรแกรมบน Arduino

(<https://i0.wp.com/www.thaieasyelec.com/images/basic-/interview-arduino-part1/3.png>)

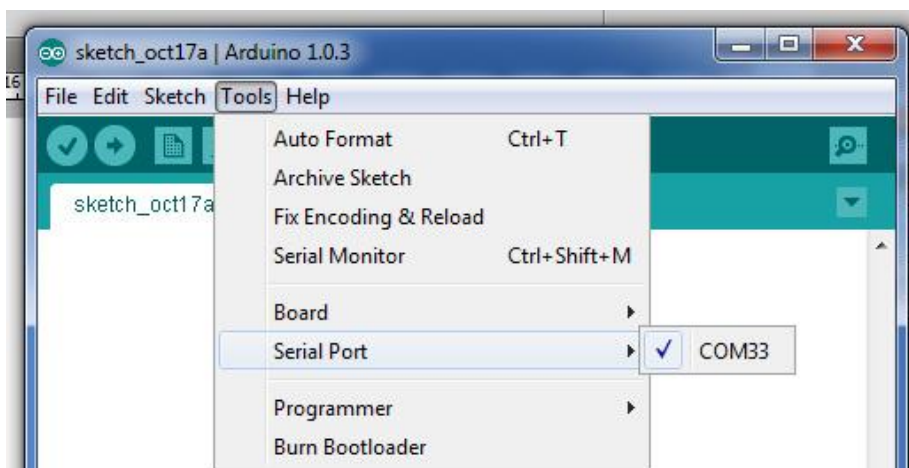
2.7.3 ขั้นตอนการอัปโหลดโค้ดลงบอร์ด Arduino

2.7.3.1 เขียนโปรแกรมบนคอมพิวเตอร์ ผ่านทางโปรแกรม ArduinoIDE ซึ่งสามารถดาวน์โหลดได้จาก [Arduino.cc/en/main/software](https://www.arduino.cc/en/main/software)

2.7.3.2 หลังจากเขียนโค้ดโปรแกรมเรียบร้อยแล้ว ให้ผู้ใช้งานเลือกบอร์ด Arduino ที่ใช้และหมายเลข Com port

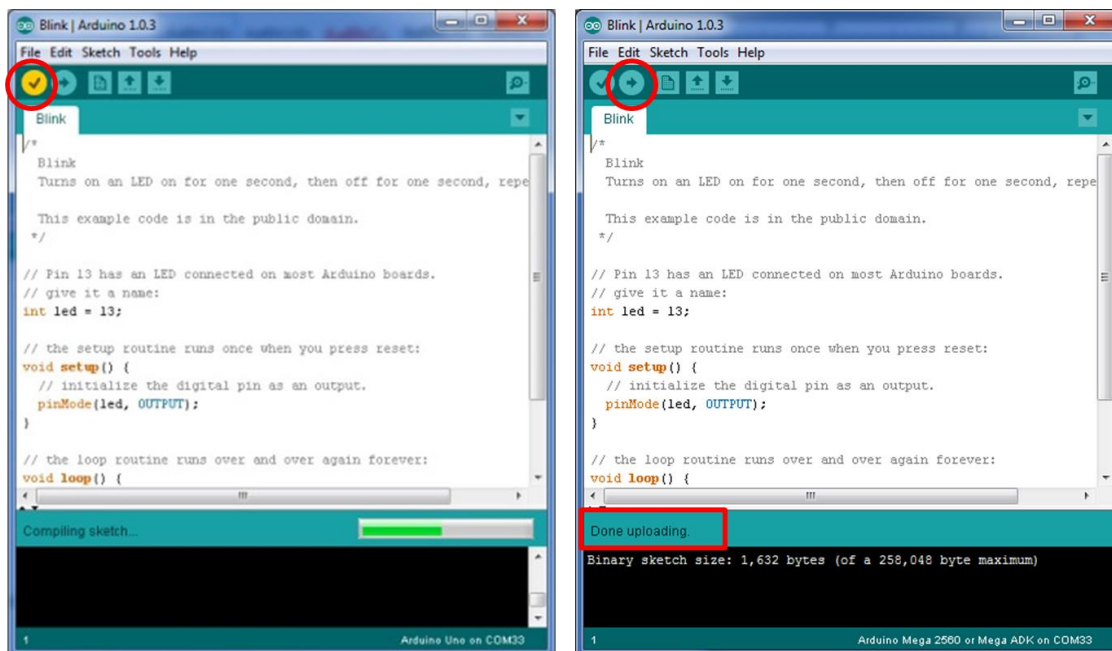


รูปที่ 2.8 เลือกบอร์ด Arduino ที่ต้องการ Upload(<https://cz.lnwfile.com/2t/yw.png>)

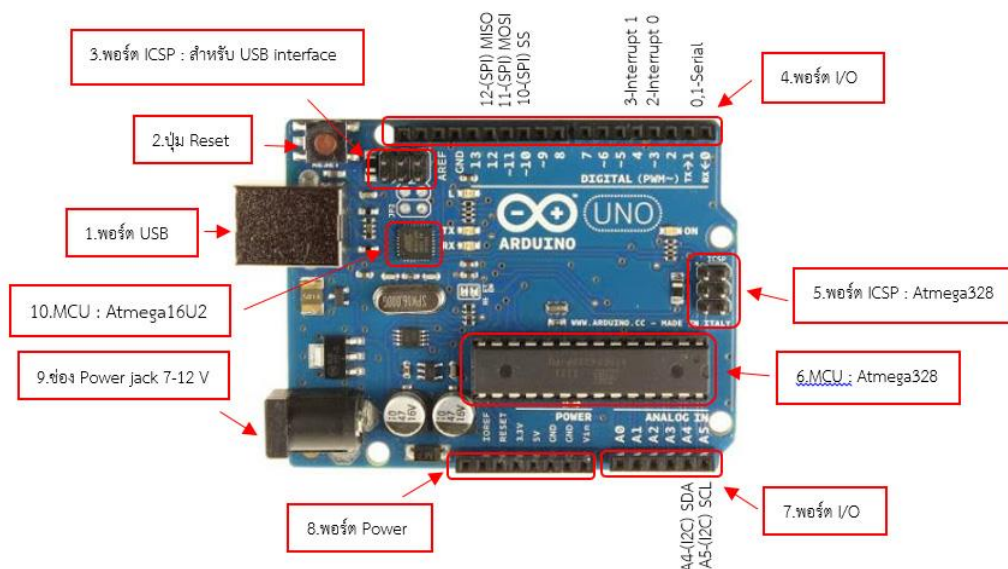


รูปที่ 2.9 เลือกหมายเลข Comport ของบอร์ด(https://cz.lnwfile.com/_/cz/_raw/o0/8a/4p.png)

กดปุ่ม Verify เพื่อตรวจสอบความถูกต้องและ Compile โค้ดโปรแกรมจากนั้นกดปุ่ม Upload โค้ด โปรแกรมไปยังบอร์ด Arduino ผ่านทางสาย USB เมื่ออัปโหลดเรียบร้อยแล้ว จะแสดงข้อความแถบข้างล่าง “Done uploading” และบอร์ดจะเริ่มทำงานตามที่เขียนโปรแกรมไว้ได้ทันที



รูปที่ 2.10 Layout & Pin out Arduino Board (http://ci.lnwfile.com/_/ci/_raw/oh/y1/w4.jpg)



รูปที่ 2.11 ข้อมูลของส่วนต่างๆในบอร์ด Arduino(https://cz.lnwfile.com/_/cz/_raw/og/5t/yq.jpg)

2.7.4 ข้อมูลของส่วนต่างๆ ในบอร์ด Arduino

2.7.4.1 USBPort: ใช้สำหรับต่อกับ Computer เพื่ออัปโหลดโปรแกรมเข้า MCU และทำไฟให้กับบอร์ด

2.7.4.2 Reset Button: เป็นปุ่ม Reset ใช้กดเมื่อต้องการให้ MCU เริ่มการทำงานใหม่

2.7.4.3 ICSP Port ของ Atmega16U2 เป็นพอร์ตที่ใช้โปรแกรม Visual Com port

2.7.4.4 I/O Port: Digital I/O ตั้งแต่ขา D0 ถึง D13 นอกจากนี้ บาง Pin จะทำหน้าที่อื่นๆ เพิ่มเติมด้วย เช่น Pin0,1 เป็นขา Tx,Rx Serial, Pin3,5,6,9,10 และ 11 เป็นขา PWM

2.7.4.5 ICSP Port: Atmega328 เป็นพอร์ตที่ใช้โปรแกรม Bootloader

2.7.4.6 MCU: Atmega328 เป็น MCU ที่ใช้บนบอร์ด Arduino

2.7.4.7 I/O Port: นอกจากจะเป็น Digital I/O แล้วยังเปลี่ยนเป็นช่องรับสัญญาณอนาล็อก ตั้งแต่ขา A0-A5

2.7.4.8 Power Port: ไฟเลี้ยงของบอร์ดเมื่อต้องการทำไฟให้กับวงจรภายนอก ประกอบด้วยขาไฟเลี้ยง +3.3 V, +5V, GND, V_{in}

2.7.4.9 Power Jack: รับไฟจาก Adapter โดยที่แรงดันอยู่ระหว่าง 7-12 V

2.7.4.10 MCU ของ Atmega16U2 เป็น MCU ที่ทำหน้าที่เป็น USB to Serial โดย Atmega328 จะติดต่อกับ Computer ผ่าน Atmega16U2

2.7.5 การทำงานของ Code

2.7.5.1 CODE ควบคุมการทำงานของ Node MCU / ESP8266

ตัวอย่างโค้ดควบคุมการทำงานของบอร์ด ESP8266

```
#include <ESP8266WiFi.h>
```

```
#include "DHT.h"
```

```
#define DHTPIN D4 // ขา Out ของ Sensor ต่อเข้าขา D4 ของ Esp8266
```

```
//เลือกชนิดของ Sensor
```

```
#define DHTTYPE DHT11 //DHT 11
```

```
//#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321
```

```
//#define DHTTYPE DHT21 // DHT 21 (AM2301)
```

```
DHT dht(DHTPIN, DHTTYPE);
```

```
const char* ssid = "SSID Wifi"; // SSID Wifi
```

```
const char* password = "Password Wifi"; // Password Wifi
```

```
const char* host = "api.thingspeak.com"; // Host ของ thingspeak ไม่ต้องแก้ไข
```

```

const char* api = "RRHS37ETW76RFAWB"; //API Key ที่เราจำไว้ ในขั้นตอน
เมื่อ
void setup() {
  Serial.begin(9600);
  delay(10);
  // We start by connecting to a WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  Serial.println("DHTxx test!");
  dht.begin();
}

int value = 0;
void loop() {
  delay(5000);
  ++value;
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  float f = dht.readTemperature(true);
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
}

```

```

}

float hif = dht.computeHeatIndex(f, h);
float hic = dht.computeHeatIndex(t, h, false);

Serial.print("Humidity: ");
Serial.print(h);
Serial.print(" %\t");
Serial.print("Temperature: ");
Serial.print(t);
Serial.print(" *C ");
Serial.print(f);
Serial.print(" *F\t");
Serial.print("Heat index: ");
Serial.print(hic);
Serial.print(" *C ");
Serial.print(hif);
Serial.println(" *F");
Serial.print("connecting to ");
Serial.println(host);

// Use WiFiClient class to create TCP connections
WiFiClient client;
const int httpPort = 80;
if (!client.connect(host, httpPort)) {
Serial.println("connection failed");
return;
}

// We now create a URI for the request
String url = "/update?api_key=";
url += api;
url += "&field1=";
url += t;
url += "&field2=";
url += h;

```



```
//เราจะส่งข้อมูล
https://api.thingspeak.com/update?api_key=RRHS37ETW76RFAWB&field1=
(อุณหภูมิ)&field2=(ความชื้น)
Serial.print("Requesting URL: ");
Serial.println(url);

// This will send the request to the server
client.print(String("GET ") + url + " HTTP/1.1\r\n" +
"Host: " + host + "\r\n" +
"Connection: close\r\n\r\n");
delay(10);

// Read all the lines of the reply from server and print them to Serial
while(client.available()){
String line = client.readStringUntil('\r');
Serial.print(line);
}
Serial.println();
Serial.println("closing connection");
}
```

2.7.5.2 CODE ควบคุมการทำงาน LCD Display

รายละเอียดคำสั่งในการสั่งงานระหว่าง Arduino กับ จอ LCD คำสั่งในการควบคุมจอ LCD ของ Arduino นั้น ทาง Arduino.cc เขียนเป็น Library มาให้เพื่อสะดวกในการนำไปใช้งาน หลังจากต่อสายเสร็จเรียบร้อยแล้ว ขั้นตอนแรกในการเริ่มเขียนโปรแกรมคือการเรียกใช้ Library ของ LCD จากไฟล์ชื่อ LiquidCrystal.h หลังจากนั้นมาดูกันว่าฟังก์ชันที่สำคัญอะไรบ้างที่ใช้สั่งงานให้จอ LCD ฟังก์ชัน LiquidCrystal(); ใช้ประกาศที่ต้องการส่งข้อมูลไปยังจอ LCD รูปแบบในการสั่งงานคือ LiquidCrystal lcd(rs, enable, d4, d5, d6, d7) ในกรณีใช้งานแบบ 4 บิต LiquidCrystal lcd(rs, enable, d0, d1, d2, d3, d4, d5, d6, d7) ในกรณีใช้งานแบบ 8 บิตในบทความนี้ใช้แบบ 4 บิต คือ LiquidCrystal lcd(12, 11, 4, 5, 6, 7); ก็หมายถึงการเชื่อมต่อ rs ที่ขา 12 , Enable ที่ขา 11 , และ DB4-DB7 ที่ขา 4-7 ของ Arduino ตามลำดับฟังก์ชัน begin(); ใช้กำหนดขนาดของจอใช้ขนาด 16 ตัวอักษร 2 บรรทัด จึงประกาศเป็น lcd.begin(16, 2); ฟังก์ชัน setCursor(); ใช้กำหนดตำแหน่งและบรรทัดของ Cursor เช่น lcd.setCursor(0, 1); คือ ให้เคอร์เซอร์ไปที่ตำแหน่งที่ 0 บรรทัดที่ 1 การนับตำแหน่งเริ่มจาก 0 ดังนั้น LCD 16x2 มีตำแหน่ง

0 – 15 บรรทัด คือ 0 กับ 1 ฟังก์ชัน print(); ใช้กำหนดข้อความที่ต้องการแสดง เช่น
 lcd.print("ThaiEasyElec"); คือ ให้แสดงข้อความ

ตัวอย่างโค้ด

```
#include <LiquidCrystal.h> //ประกาศLibrary

// initialize the library with the numbers of the interface pins

LiquidCrystal lcd(12, 11, 4, 5, 6, 7); //ฟังก์ชันแรกกำหนดPins ที่ใช้ในการเชื่อมต่อ

void setup()
{
  // set up the LCD's number of columns and rows:

  lcd.begin(16, 2); //กำหนดขนาดของจอ columns และ rows

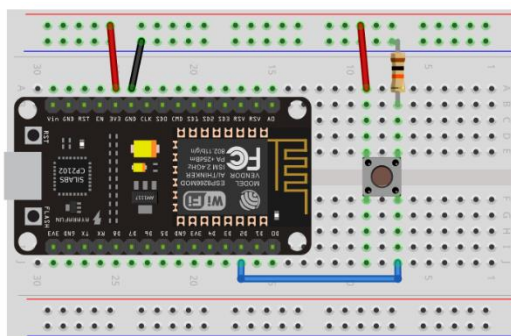
  lcd.print("Hello !!!"); //กำหนดข้อความที่ต้องการแสดงผล

  lcd.setCursor(0, 1); //กำหนดตำแหน่ง Cursor

  lcd.print("ThaiEasyElec"); //กำหนดข้อความที่ต้องการแสดงผล

  void loop()}
```

2.7.5.3 การต่อวงจรสวิตช์ให้ NodeMCU v1.0 ในบทความนี้จะเลือกใช้ NodeMCU เนื่องจากเป็นบอร์ดที่ค่อนข้างได้รับความนิยมเป็นส่วนใหญ่ ในการทดลอง จะใช้สวิตช์ ในการแทนเซ็นเซอร์อื่น ๆ เมื่อมีการกดสวิตช์แล้ว จะมีการส่งแจ้งเตือนไปทาง LINE ท่านสามารถ เปลี่ยนสวิตช์เป็น PIR Sensor เพื่อตรวจจับผู้บุกรุกได้ หรือเปลี่ยนเป็นสวิตช์แม่เหล็กติดกับประตู เพื่อแจ้งเตือนมีการเปิดประตูได้



รูปที่ 2.12 วงจรสวิตช์ NodeMCU v.10(https://cu.lnwfile.com/_/cu/_raw/me/x9/dx.jpg)

Codingในโค้ดด้านล่างนี้ ท่านสามารถคัดลอกไปวางในโปรแกรม Arduino IDE ได้เลย แล้วแก้ไข <WIFI_SSID> , <WIFI_PASSWORD> ให้เป็นค่าที่ถูกต้อง ส่วน <LINE_TOKEN> ให้ นำ Access Token จากในขั้นตอนที่แล้วมาวาง

```
void Line_Notify(String message) ;
```

```

#include <ESP8266WiFi.h>

// Config connect WiFi

#define WIFI_SSID "<YOUR WIFINAME>"

#define WIFI_PASSWORD "<YOUR WIFIPASSWORD>"

// Line config

#define LINE_TOKEN "<LINE ACCESS TOKEN>"

#define SW D2

String message = PLEASE INSERT COIN";

void setup() {
  pinMode(SW, INPUT);
  Serial.begin(9600);
  WiFi.mode(WIFI_STA);
  // connect to wifi.
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());
}

void loop() {
  if (digitalRead(SW) == HIGH) {
    while(digitalRead(SW) == HIGH) delay(10);
    Serial.println("Enter !");
    Line_Notify(message);
    // Serial.println();
  }
  delay(10);
}

```

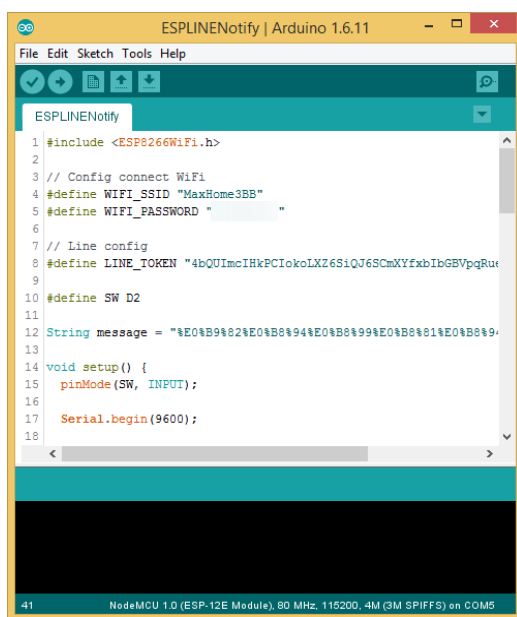
```

void Line_Notify(String message) {
  WiFiClientSecure client;

  if (!client.connect("notify-api.line.me", 443)) {
    Serial.println("connection failed");
    return;
  }
  String req = "";
  req += "POST /api/notify HTTP/1.1\r\n";
  req += "Host: notify-api.line.me\r\n";
  req += "Authorization: Bearer " + String(LINE_TOKEN) + "\r\n";
  req += "Cache-Control: no-cache\r\n";
  req += "User-Agent: ESP8266\r\n";
  req += "Content-Type: application/x-www-form-urlencoded\r\n";
  req += "Content-Length: " + String(String("message=" + message).length()) +
  "\r\n";
  req += "\r\n";
  req += "message=" + message;
  // Serial.println(req);
  client.print(req);
  delay(20);
  // Serial.println("-----");
  while(client.connected()) {
    String line = client.readStringUntil('\n');
    if (line == "\r") {
      break;
    }
    //Serial.println(line);
  }
  // Serial.println("-----");
}

```

2.7.5.4 จากนั้นอัปโหลดลง NodeMCU ไปได้เลย



รูปที่ 2.13 การอัปโหลดโค้ดลง NodeMCU()https://cu.lnwfile.com/_/cu/_raw/me/b10/dx.jpg

บทที่ 3

วิธีการดำเนินโครงการ

ในการดำเนินการเขียนชุดคำสั่งควบคุมมอเตอร์ไต่เก็บโน้ตบุ๊กอิเล็กทรอนิกส์ มีขั้นตอนการเขียนต่างๆ โดยทางคณะผู้จัดทำได้ร่วมกันวางแผนในการปฏิบัติงาน และจัดแบ่งงานขั้นตอนการดำเนินการสร้างชิ้นงานตามความเหมาะสม

ขั้นตอนในการดำเนินการสร้างชิ้นงาน แบ่งออกเป็นดังนี้

- 3.1 วางแผนการเขียนชุดคำสั่ง
- 3.2 ออกแบบการทำงานของระบบ
- 3.3 ดำเนินการเขียนชุดคำสั่ง

3.1 การวางแผนการเขียนชุดคำสั่ง

การวางแผนและการเตรียมการ เริ่มเมื่อคณะกรรมการพิจารณาโครงการให้เสนอหัวข้อโครงการในภาคเรียนที่ 1 ทางคณะผู้จัดทำได้เสนอหัวข้อโครงการ ชุดคำสั่งควบคุมมอเตอร์ไต่เก็บโน้ตบุ๊กอิเล็กทรอนิกส์

ซึ่งมีลำดับขั้นตอนต่างๆ ในการดำเนินโครงการดังตารางที่ 3.1

3.1.1 การวางแผนทำโครงการ

- 1) เตรียมเสนอหัวข้อโครงการกับคณะกรรมการ
- 2) หาข้อมูลเกี่ยวกับคำสั่ง Arduino
- 3) เขียนโครงการบทที่ 1
- 4) ได้รับการอนุมัติโครงการจากคณะกรรมการ
- 5) ศึกษาเกี่ยวกับการทำงานมอเตอร์
- 6) วาด FlowChart การทำงานของระบบ
- 7) เตรียมอุปกรณ์ที่ใช้ทดลอง มอเตอร์ สายไฟ สวิตช์ บอร์ด
- 8) เริ่มเขียนชุดคำสั่ง
- 9) ต่ออุปกรณ์ต่างๆ ที่ใช้ทดลอง
- 10) ทดสอบระบบการทำงาน
- 11) แก้ไขข้อผิดพลาด
- 12) นำโค้ดที่สมบูรณ์ไปใช้ร่วมกับไต่
- 13) ทดสอบการทำงาน
- 14) แก้ไขข้อผิดพลาด

- 15) เขียนโครงการบทที่ 2-5 โดยนำข้อมูลจากการสร้างชิ้นงาน , การออกแบบชิ้นงาน , ผลของการทำงาน , ข้อมูลต่างๆ ที่ได้ มาเรียบเรียงให้ได้ ใจความที่สมบูรณ์
- 16) ส่งโครงการบทที่ 1-5 โดยให้ที่ปรึกษาโครงการตรวจสอบ
- 17) ยื่นขอสอบโครงการ หลังจากผ่านบทที่ 1-5 ที่ผ่านการตรวจสอบความถูกต้อง แล้ว
- 18) อนุมัติสอบโครงการ
- 19) สอบโครงการบทที่ 1-5 เป็นการนำเสนอเนื้อหาต่างๆ กับคณะกรรมการที่สอบโครงการ ให้ทราบข้อมูลต่างๆ ที่เกี่ยวข้องกับการทำโครงการนี้
- 20) ส่งโครงการโดยนำเนื้อหาบทที่ 1-5 มาเข้าเล่ม แล้วนำไปให้กับคณะกรรมการสอบโครง การเป็นหลักฐานในการศึกษา

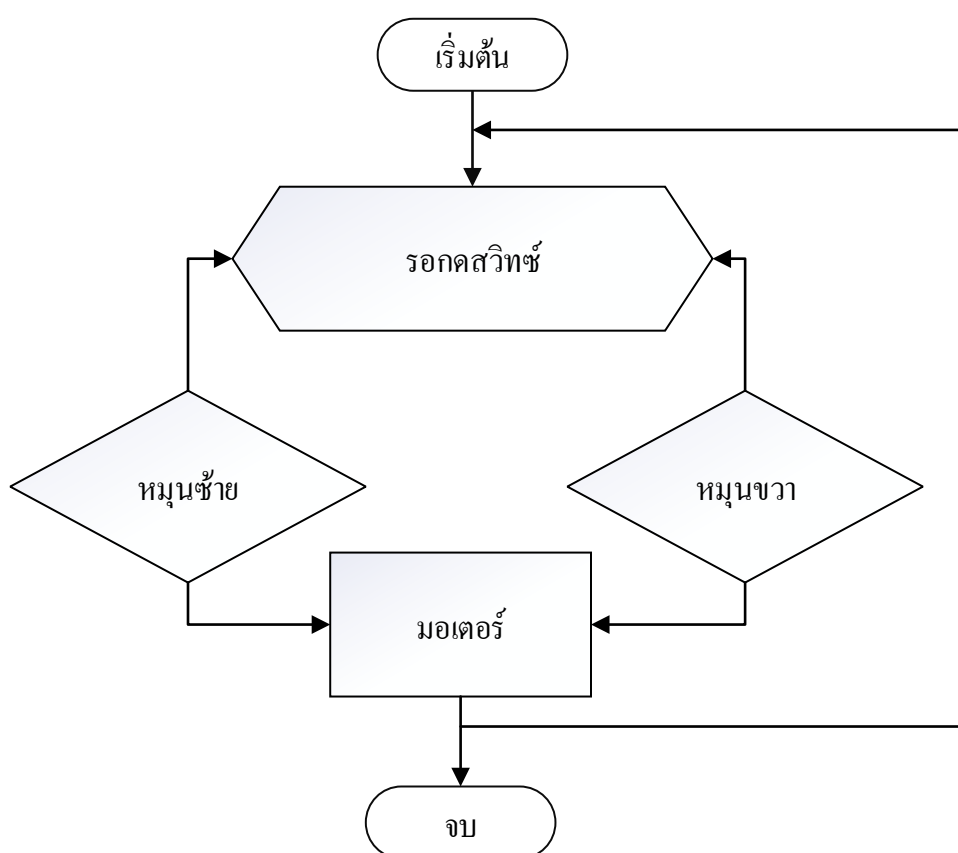
3.2 ออกแบบการทำงานของระบบ

การออกแบบผังการทำงานของชุดคำสั่ง

ในการเขียนชุดคำสั่ง จะต้องมีการเขียนรูปแบบการทำงานของคำสั่งก่อน ซึ่งมีหลักการทำงานดังนี้

1) รูปแบบการทำงานของชุดคำสั่ง

ในการทำงาน จะต้องกดปุ่มสวิตช์เพื่อทำงาน โดยจะแยกออกเป็น 2 เงื่อนไขคือ ให้หมุนซ้าย หรือ หมุนขวา หากผ่านเงื่อนไขใดเงื่อนไขหนึ่ง ก็จะให้มอเตอร์ทำงานตามเงื่อนไขนั้น เช่น หาก กดปุ่ม ครั้งที่ 1 ก็จะหมุนซ้าย หากกดครั้งที่ 2 ก็จะหมุนขวา



รูปที่ 3.1 แสดง Flowchart รูปแบบการทำงานของชุดคำสั่ง

3.3 ดำเนินการเขียนชุดคำสั่ง

ในการเขียนคำสั่ง จะเขียนผ่านโปรแกรมซอฟต์แวร์ที่ชื่อว่า Arduino IDE คือ โปรแกรมสำหรับ คอมไพล์ และอัปโหลดโปรแกรมลง บอร์ด Arduino หรือบอร์ดตัวอื่นๆ ที่คล้ายกัน โดยชุดคำสั่งที่ใช้งานในครั้งนี้ จะมีรูปแบบการทำงานดังนี้

```
int motor1 = 2;
int motor2 = 4;
int motor3 = 5;
int motor4 = 6;

int buttonPin = 3;
int buttonStart;
int Menu = 0;
int lastinput=HIGH;

void setup(){
  pinMode(motor1, OUTPUT);
  pinMode(motor2, OUTPUT);

  pinMode(buttonPin, INPUT) ;
  Serial.begin(9600);
}

void loop()

{
  buttonStart = digitalRead(buttonPin);
  if (buttonStart == lastinput)
  {
    delay (300);
    Menu++;
    if (Menu>2)
      Serial.print(Menu);
```

```
if (Menu==1)
{
    Serial.println("Motor Left");

    digitalWrite(motor1,LOW);
    digitalWrite(motor1, HIGH);
    delay (4000);
    digitalWrite(motor1,LOW);

    digitalWrite(motor3,LOW);
    digitalWrite(motor3, HIGH);
    delay (5000);
    digitalWrite(motor3,LOW);
    lastinput=LOW;
}

else if(Menu==2)
{
    Serial.println("Motor Right");

    digitalWrite(motor4,LOW);
    digitalWrite(motor4, HIGH);
    delay (5000);
    digitalWrite(motor4,LOW);

    digitalWrite(motor2,LOW);
    digitalWrite(motor2, HIGH);
    delay (4000);

    digitalWrite(motor2,LOW);
    lastinput=HIGH;
```

```
Menu=0;  
}  
}  
}  
digitalWrite(motor2,LOW);  
Menu=0;  
}  
}  
}
```

บทที่ 4

ผลการศึกษา

ผลการศึกษาทำให้เราได้รู้ถึงการทำงานร่วมกันเป็นกลุ่มต้องสามัคคีกันในการวิเคราะห์ และออกแบบตลอดจนถึงขั้นตอนการทำงานแต่ละขั้นตอนว่ามีอะไรบ้าง ในที่นี้ทางคณะผู้จัดทำจะกล่าวถึงผลของการศึกษาข้อมูล และผลที่ได้รับอย่างละเอียด แบ่งออกเป็นดังนี้

- 4.1 ขั้นตอนการวางแผนการเขียนชุดคำสั่ง
- 4.2 ขั้นตอนการดำเนินการเขียนชุดคำสั่ง
- 4.3 ขั้นตอนการทดสอบ
- 4.4 ผลการศึกษา

4.1 ขั้นตอนการวางแผนการเขียนชุดคำสั่ง

ขั้นตอนนี้คณะผู้จัดทำได้ทำการประชุมกลุ่มกันออกความคิดเห็นและเพื่อแบ่งหน้าที่ของแต่ละคนเพื่อให้ขั้นตอนในการทำงานทำงานได้อย่างรวดเร็ว และมีการกระจายหน้าที่การทำงานให้เท่าเทียมกันเพื่อให้ทุกคนได้แสดงความคิดเห็นได้อย่างอิสระ และให้ทุกคนมีส่วนร่วมในการทำงานและรับความคิดเห็นของผู้อื่นด้วย ขั้นตอนการออกแบบมีดังนี้

- 4.1.1 หาข้อมูลเกี่ยวกับคำสั่ง Arduino
- 4.1.2 ศึกษาเกี่ยวกับการทำงานมอเตอร์
- 4.1.3 วาด FlowChart การทำงานของระบบ
- 4.1.4 เตรียมอุปกรณ์ที่ใช้ทดลอง มอเตอร์ สายไฟ สวิตช์ บอร์ด
- 4.1.5 เริ่มเขียนชุดคำสั่ง
- 4.1.6 ต่ออุปกรณ์ต่างๆ ที่ใช้ทดลอง
- 4.1.7 ทดสอบระบบการทำงาน
- 4.1.8 นำCodeที่สมบูรณ์ไปใช้ร่วมกับโต๊ะ
- 4.1.9 ทดสอบการทำงาน
- 4.1.10 แก้ไขข้อผิดพลาด

4.2 ขั้นตอนการดำเนินการเขียนชุดคำสั่ง

- 4.2.1 ศึกษาการเขียนโปรแกรม
- 4.2.2 เขียนโปรแกรมซอฟต์แวร์ Arduino IDE
- 4.2.3 อัปโหลดโปรแกรมลง บอร์ด Arduino IDE หรือบอร์ดตัวอื่นๆ ที่คล้ายกัน

4.3 ขั้นตอนการทดสอบ

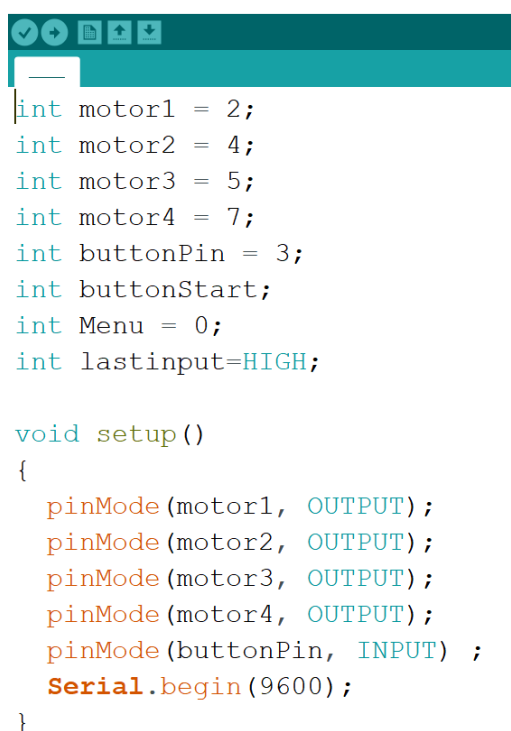
เราได้ทำการออกแบบให้ผู้ใช้กดปุ่มสวิตช์สัมผัสในการใช้งานเปิดและปิดตัวขึ้นงานในการใส่โน้ตบุ๊กเพื่อจัดเก็บ หรือนำโน้ตบุ๊กออกมาใช้งาน เพื่อความสะดวกในการใช้งานของผู้ใช้งาน ได้ทำการทดสอบดังนี้

4.3.1 ทดสอบส่วนประกอบเทคนิค

4.3.2 ทดสอบเสถียรภาพในการทำงาน

4.4 ผลการศึกษา


ในการทำงาน จะต้องกดปุ่มสวิตช์เพื่อทำงาน โดยจะแยกออกเป็น 2 เงื่อนไขคือ ให้หมุนซ้าย หรือ หมุนขวา หากผ่านเงื่อนไขใดเงื่อนไขหนึ่ง ก็จะให้มอเตอร์ทำงานตามเงื่อนไขนั้น เช่น หาก กดปุ่ม ครั้งที่ 1 ก็จะหมุนซ้าย หากกดครั้งที่ 2 ก็จะหมุนขวา โดยชุดคำสั่งที่ใช้งานในครั้งนี้ จะมีรูปแบบการทำงานดังนี้



```
int motor1 = 2;
int motor2 = 4;
int motor3 = 5;
int motor4 = 7;
int buttonPin = 3;
int buttonStart;
int Menu = 0;
int lastinput=HIGH;

void setup()
{
  pinMode(motor1, OUTPUT);
  pinMode(motor2, OUTPUT);
  pinMode(motor3, OUTPUT);
  pinMode(motor4, OUTPUT);
  pinMode(buttonPin, INPUT);
  Serial.begin(9600);
}
```

รูปที่ 4.1 การประกาศตัวแปร และการ Setup IN/OUT




```

void loop()
{

    buttonStart = digitalRead(buttonPin);
    if (buttonStart == lastinput)
    {
        delay (300);
        Menu++;
        if (Menu>2)
            Serial.print(Menu);
    }
}

```

รูปที่ 4.2 การทำงานของปุ่มกดในรูปแบบ loop



```


if (Menu==1)
{
    Serial.println("Motor Left");

    digitalWrite(motor1,LOW);
    digitalWrite(motor1, HIGH);
    delay (4230);
    digitalWrite(motor1,LOW);

    digitalWrite(motor3,LOW);
    digitalWrite(motor3, HIGH);
    delay (4500);
    digitalWrite(motor3,LOW);
    lastinput=LOW;
}

```

รูปที่ 4.3 การทำงานของมอเตอร์ (เปิดช่องโน้ตบุ๊ก)



```

else if(Menu==2)
{
    Serial.println("Motor Right");

    digitalWrite(motor4,LOW);
    digitalWrite(motor4, HIGH);
    delay (3550);
    digitalWrite(motor4,LOW);

    digitalWrite(motor2,LOW);
    digitalWrite(motor2, HIGH);
    delay (4200);
    digitalWrite(motor2,LOW);
    lastinput=HIGH;
    Menu=0;
}
}
}

```

รูปที่ 4.4 การทำงานของมอเตอร์ (ปิดช่องโน้ตบุ๊ก)

บทที่ 5

สรุป อภิปรายผล และข้อเสนอแนะ

ในการดำเนินการเขียนชุดคำสั่งควบคุมมอเตอร์ โด๊ะเก็บโน้ตบุ๊กอิเล็กทรอนิกส์ นั้นสามารถบรรลุวัตถุประสงค์ที่วางไว้ ซึ่งอุปกรณ์สามารถทำงานได้อย่างที่ต้องการ แต่การดำเนินโครงการก็ประสบปัญหาต่างๆ หลายอย่าง ซึ่งผู้ดำเนินโครงการมีข้อเสนอแนะที่จะนำมาใช้พัฒนาปรับปรุงแก้ไขให้ชุดคำสั่งควบคุมมอเตอร์ โด๊ะเก็บโน้ตบุ๊กอิเล็กทรอนิกส์มีประสิทธิภาพการทำงานของอุปกรณ์ดีที่สุด

5.1 วัตถุประสงค์ของโครงการ

- 5.1.1 เพื่อสร้างโด๊ะสำหรับเก็บโน้ตบุ๊กใช้งานในพื้นที่จำกัด
- 5.1.2 เพื่อจัดเก็บ Notebook ได้อย่างปลอดภัย
- 5.1.3 เพื่อนำความรู้ที่ได้จากการเรียนการสอนในสาขาวิชาเทคโนโลยีสารสนเทศมาประยุกต์ใช้ในการทำโครงการ

5.2 ประโยชน์ที่ได้รับจากการศึกษา

- 5.2.1 ได้โด๊ะญี่ปุ่นใช้งานในพื้นที่จำกัด
- 5.2.2 จัดเก็บ Notebook ได้อย่างปลอดภัย
- 5.2.3 เพื่อนำความรู้ที่ได้จากการเรียนการสอนในสาขาวิชาเทคโนโลยีสารสนเทศมาประยุกต์ใช้ในการทำโครงการ

5.3 ปัญหาที่ประสบในการดำเนินโครงการ

ในการดำเนินการเขียนชุดคำสั่งควบคุมมอเตอร์ โด๊ะเก็บโน้ตบุ๊กอิเล็กทรอนิกส์นั้นทางคณะผู้จัดทำได้ประสบปัญหาการดำเนินโครงการหลายอย่างในที่นี้ทางคณะผู้จัดทำโครงการจะอธิบายสาเหตุและวิธีการแก้ปัญหาเป็นข้อ ๆ ดังนี้

- 5.3.1 ปัญหาด้านการศึกษาชุดคำสั่ง เนื่องจากชุดคำสั่งที่ซับซ้อน คณะผู้จัดทำจึงจำเป็นต้องศึกษาข้อมูลจากหลายๆแหล่งเพื่อให้ทราบถึงการใช้ชุดคำสั่งต่าง
- 5.3.2 ปัญหาด้านการทำงานของโด๊ะเก็บโน้ตบุ๊กอัตโนมัติ เนื่องจากเกิดความผิดพลาดจากเรื่องระยะเวลาในการเปิดหรือปิด และแรงของมอเตอร์ น้ำหนักของโน้ตบุ๊กทำให้เกิดการคลาดเคลื่อน
- 5.3.3 ปัญหาด้านวงจรสายไฟเชื่อมต่อกันไม่แน่นหรือหลุดทำให้มอเตอร์ไม่ทำงาน จึงทำให้ระบบเกิดการขัดข้อง

5.4 ผลการดำเนินโครงการ

ในการดำเนินโครงการชุดคำสั่งควบคุมมอเตอร์ โต้ะเก็บน้ำอัตโนมัติอิเล็กทรอนิกส์เริ่มตั้งแต่การเสนอโครงการต่อคณะกรรมการพิจารณา ทางคณะกรรมการได้ให้หาข้อมูลเพิ่มเติมเรียบร้อย จึงได้รับการอนุมัติจากคณะกรรมการในการจัดสร้างแล้วมาการศึกษาข้อมูลรายละเอียดต่างๆ เกี่ยวกับการดำเนินชุดคำสั่งควบคุมมอเตอร์ โต้ะเก็บน้ำอัตโนมัติอิเล็กทรอนิกส์โดยได้ทำการออกแบบและดำเนินการจัดทำตามที่วางแผนไว้จนสำเร็จ

ผลการดำเนินโครงการชุดคำสั่งควบคุมมอเตอร์ โต้ะเก็บน้ำอัตโนมัติอิเล็กทรอนิกส์นี้ซึ่งสามารถทำงานระบบได้สำเร็จ และช่วยให้ ทำงานกับตัวโต้ะเก็บน้ำอัตโนมัติอิเล็กทรอนิกส์ได้

5.5 อภิปรายผล

จากผลของการดำเนินโครงการนี้ถือว่าประสบความสำเร็จตามที่ตั้งจุดประสงค์ไว้ คือสามารถเก็บน้ำอัตโนมัติได้ปกติ นอกจากนี้คณะผู้จัดทำยังได้รับความรู้และประสบการณ์ในการทำโครงการนี้เป็นอย่างมาก การจัดทำโครงการครั้งนี้ ทางกลุ่มผู้จัดทำโครงการได้ศึกษาข้อมูลจากเอกสารและผลงานที่เกี่ยวข้องกับงานวิจัยจาก เว็บไซต์ My Arduino ซึ่งสอนการเขียนชุดคำสั่งต่างมากมาย สอนใช้งาน Arduino ควบคุมมอเตอร์ DC หมุนซ้าย หมุนขวา โดยใช้ตัวขับเคลื่อนมอเตอร์ L298N ในการควบคุมมอเตอร์ โดยมีคลิปวิดีโอสื่อการสอน ขั้นตอนการต่อวงจรที่เข้าใจง่ายโดยสรุปมาแล้วและยังมีข้อมูลสอนการใช้ Arduino อีกมากมาย

5.6 ข้อเสนอแนะ

5.6.1 ข้อเสนอทั่วไป

5.6.1.1 แนะนำในการเพิ่มช่องที่เก็บน้ำ

5.6.1.2 ควรมีน้ำหนักที่เบาและสามารถเคลื่อนย้ายได้สะดวก

5.6.2 ข้อเสนอแนะทางเทคนิค

5.6.2.1 ตัวเครื่องควรมีระบบรักษาความปลอดภัยมากกว่านี้ เช่น กุญแจล็อกช่องที่เก็บ Notebook เครื่องไว้

บรรณานุกรม

ทีมงานพัฒนาเว็บไซต์ Thaiall. (2560). **ผังงาน(Flowchart)**. ค้นข้อมูล 1 มิถุนายน 2562, จาก

<http://www.thaiall.com/flowchart/indexo.html>

บริษัท วินัส ซัพพลาย จำกัด. (2560). **บทความ Arduino คืออะไร ตอนที่1 แนะนำเพื่อนใหม่ที่ชื่อ**

Arduino. ค้นข้อมูล 1 มิถุนายน 2562, จาก [http://www.thaicasylec.com/article-wiki/latest-](http://www.thaicasylec.com/article-wiki/latest-blogs/what-is-arduino-ch1.html)

[blogs/what-is-arduino-ch1.html](http://www.thaicasylec.com/article-wiki/latest-blogs/what-is-arduino-ch1.html)

วิกิพีเดียสารานุกรมเสรี. (2562). **ภาษาซี (C Programming Language)**. ค้นข้อมูล 1 มิถุนายน

2562, จาก [https:// th.wikipedia.org/wiki/ภาษาซี](https://th.wikipedia.org/wiki/ภาษาซี)

เจ้าของร้าน. (2559). **Arduino คืออะไร**. ค้นข้อมูล 1 มิถุนายน 2562, จาก [http://www.ioxhop.com/](http://www.ioxhop.com/article/1/arduino-ตอนที่1-arduino-คืออะไร)

[article/1/arduino-ตอนที่1-arduino-คืออะไร](http://www.ioxhop.com/article/1/arduino-ตอนที่1-arduino-คืออะไร)

เจ้าของร้าน. (2562). **สอนใช้งาน Arduino ควบคุมมอเตอร์ DC หมุนซ้าย หมุนขวา**. ค้นข้อมูล

1 มิถุนายน 2562, จาก [https://www.myarduino.net/article/101/สอนใช้งาน-arduino-ควบคุม](https://www.myarduino.net/article/101/สอนใช้งาน-arduino-ควบคุมมอเตอร์-dc-หมุนซ้าย-หมุนขวา)

[มอเตอร์-dc-หมุนซ้าย-หมุนขวา](https://www.myarduino.net/article/101/สอนใช้งาน-arduino-ควบคุมมอเตอร์-dc-หมุนซ้าย-หมุนขวา)

เจ้าของร้าน. (2562). **สอนใช้งาน Arduino if else ตรวจสอบเงื่อนไข ถูกผิด**. ค้นข้อมูล 1 มิถุนายน

2562, จาก [https://www.myarduino.net/article/87/สอนใช้งาน-arduino-if-else-ตรวจสอบ](https://www.myarduino.net/article/87/สอนใช้งาน-arduino-if-else-ตรวจสอบเงื่อนไข-ถูกผิด)

[เงื่อนไข-ถูกผิด](https://www.myarduino.net/article/87/สอนใช้งาน-arduino-if-else-ตรวจสอบเงื่อนไข-ถูกผิด)

ภาคผนวก ก
แบบเสนอร่างโครงการ

ภาคผนวก ข

ความคืบหน้าโครงการ

ภาคผนวก ก

คู่มือการใช้งาน

คู่มือการใช้งาน

1. กดปุ่มสวิตช์สัมผัสที่ข้างกล่องใต้โต๊ะด้านซ้าย
2. เมื่อกดปุ่มไฟที่ปุ่มจะขึ้น
3. รรระบบทำงาน ที่ปิดช่องโน้ตบุ๊กจะค่อยๆเปิด และเลื่อนฐานวางโน้ตบุ๊กขึ้นมา
4. หากกดปุ่มอีกครั้ง ไฟที่ปุ่มจะดับ
5. รรระบบทำงาน ฐานรองโน้ตบุ๊กจะค่อยๆเก็บลงและปิดช่องเก็บโน้ตบุ๊ก



ภาคผนวก ง

ประวัติผู้จัดทำ

ประวัติผู้วิจัย



ชื่อ-ชื่อสกุล

นายภาณุพล สอนน่วม

วันเดือนปีเกิด

2 เมษายน 2545

สถานที่เกิด

สมุทรปราการ

สถานที่อยู่อยู่ปัจจุบัน

188/1 ม.2 ต.บางปูใหม่ อ.เมือง จ.สมุทรปราการ 10280

ประวัติการศึกษา

พ.ศ. 2560

ประกาศนียบัตรวิชาชีพ

สาขาวิชาเทคโนโลยีสารสนเทศ

วิทยาลัยเทคโนโลยีอรรณพวิทย์พัฒนวิชาการ

ผลงานและกิจกรรม

พ.ศ. 2562

- ได้รับรางวัลชนะเลิศการตอบคำถามภาษา Python

พ.ศ. 2561

- อบรมไทยซัมมิต รุ่นที่ 1

ประวัติผู้วิจัย



ชื่อ-ชื่อสกุล

นางสาวนางสาวอุรวรินทร์ เศรษฐนิโรจน์

วันเดือนปีเกิด

25 พฤษภาคม 2545

สถานที่เกิด

กรุงเทพมหานคร

สถานที่อยู่ปัจจุบัน

3394 ถนนสุขุมวิท แขวงบางนา เขตบางนา

กรุงเทพมหานคร 10260

ประวัติการศึกษา

พ.ศ. 2560

ประกาศนียบัตรวิชาชีพ

สาขาวิชาเทคโนโลยีสารสนเทศ

วิทยาลัยเทคโนโลยีอรรณพวิทยพัฒน์วิชาการ

ผลงานและกิจกรรม

-

ประวัติผู้วิจัย



ชื่อ-ชื่อสกุล

นางสาวรัตนภรณ์ ทองล้ำ

วันเดือนปีเกิด

20 พฤษภาคม 2544

สถานที่เกิด

กรุงเทพมหานคร

สถานที่อยู่ปัจจุบัน

99 ซ.48 ถนนบางนา-ตราด แขวงบางนา เขตบางนา
กรุงเทพมหานคร 10260

ประวัติการศึกษา

พ.ศ. 2560

ประกาศนียบัตรวิชาชีพ

สาขาวิชาเทคโนโลยีสารสนเทศ

วิทยาลัยเทคโนโลยีอรรถวิทย์พัฒนวิชาการ

ผลงานและกิจกรรม

-