**บทที่ 2**

**ทฤษฎีและหลักการที่เกี่ยวข้อง**

โครงงาน “หุ่นยนต์ G-10” ได้จัดทำขึ้นเพื่อออกแบบและจำลองวงจรที่ใช้ในการควบคุมหุ่นยนต์ โดยแบ่งกลยุทธ์ออกเป็น 2 กลยุทธ์ คือกลยุทธ์รุกและกลยุทธ์รับ ที่ควบคุมการทำงานด้วยไมโครคอนโทรลเลอร์ จึงมีทฤษฎีและหลักการที่เกี่ยวข้องดังต่อไปนี้

**2.1 ภาษา C/C++**

ภาษา Arduino (หรือ ภาษา C/C++) ใช้สำหรับเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์

Arduino IDE เป็นเครื่องมือสำหรับเขียนโปรแกรมด้วยภาษา Arduino คอมไพล์โปรแกรม (Compile) และอัปโหลดโปรแกรมลงบอร์ด (Upload)

1) ฟังก์ชันหลัก (Structure)

เป็นฟังก์ชันหลักในการเขียนโปรแกรม จำเป็นต้องมีในทุกโปรแกรม

Setup () คือ ฟังก์ชันใช้ในการประกาศค่าเริ่มต้น ตำแหน่งพอร์ตที่ใช้งาน รวมถึงฟังก์ชันที่อยู่ไลบารีที่ใช้งาน เป็นฟังก์ชันที่ทำงานเพียงครั้งเดียว จะทำงานทุกครั้งที่มีการรีเซต หรือรีบูตเครื่องใหม่เท่านั้น

Loop () คือ ฟังก์ชันใช้ในการเขียนโค้ดโปรแกรมการทำงานของ Arduino เป็นฟังก์ชันการวนลูปไปเรื่อยๆ

2) ชุดคำสั่งในการควบคุม (Control Structures)

เป็นชุดคำสั่งในการใช้ในการตัดสินใจหาทางออก เพื่อใช้ในการทำงาน

If คือ คำสั่งในการตัดสินใจแบบตัวเลือกเดียว โดยใช้งานร่วมกับ and, or not, ==, !=, <, > เพื่อใช้ในการตัดสินใจในการหาคำตอบ

If...else คือ คำสั่งในการตัดสินใจแบบหลายตัวเลือก โดยใช้งานร่วมกับ and, or not, ==, !=, <, >เพื่อใช้ในการตัดสินใจในการหาคำตอบ

for คือ คำสั่งกำหนดเงื่อนไขเป็นจำนวนครั้งที่จะทำตามชุดคำสั่งต่าง ๆ ภายในลูป เหมาะที่จะใช้กับงานประเภทที่ไม่มีการเปลี่ยนแปลง

switch case คือ คำสั่งที่ใช้ในการจัดการเงื่อนไขหลายเงื่อนไข โดยเฉพาะการใช้งานโครงสร้าง การจำแนกเงื่อนไขไม่จำเป็นต้องอาศัยเฉพาะตัวแปรที่เก็บค่าจำนวนเต็มเท่านั้น ข้อมูลแบบอื่นก็สามารถใช้ได้เช่นกัน

while คือ คำสั่งเงื่อนไขที่จะทำการตรวจสอบว่าเป็นจริงหรือเท็จ ชุดคำสั่งก็คือ ส่วนที่ทำงานซ้ำ ๆ โดยจะต้องมีคำสั่งที่จะทำให้เงื่อนไขเป็นเท็จด้วย

#define คือ คำสั่งกำหนดค่านิพจน์ต่าง ๆ ให้กับชื่อของตัวคงที่

#include การกำหนดชื่อไฟล์ตามหลัง include จะใช้เครื่องหมาย <> ซึ่งจะเป็นการอ่านไฟล์จาก ไดเร็กทอรี หรือโฟลเดอร์ที่กำหนดไว้ก่อนแล้ว โดยปกติจะเป็นโฟลเดอร์ include แต่ถ้าใช้เครื่องหมาย “ ” เป็นการอ่านไฟล์จาก โฟลเดอร์ หรือไดเร็กทอรี ที่กำลังติดต่ออยู่และไฟล์ที่จะ include เข้ามานี้จะต้องไม่มีฟังก์ชัน main () โดยมากจะประกอบไปด้วยโปรแกรมย่อย ค่าคงที่ หรือข้อความต่าง ๆ

3) Conversion การแปลงค่า

char () คือ การแปลงค่าข้อมูลให้เป็น character

byte () คือ การแปลงค่าข้อมูลให้เป็น byte

int () คือ การแปลงค่าข้อมูลให้เป็น integer

4) Time

delay () คือ คำสั่งหยุดการทำงานโปรแกรมสำหรับจำนวนของเวลา (ใน milliseconds) Milliseconds = จำนวนมิลลิวินาทีในการหยุดการทำงานชั่วคราว

2.2.5 Functions

pinMode () ใช้ในกลุ่ม void setup () เพื่อกำหนดหน้าที่ขาของไมโครคอนโทรลเลอร์ให้เป็น ขารับสัญญาณ INPUT หรือขาส่งสัญญาณ OUTPUT

digitalWrite () คือ การส่งค่าลอจิก HIGH หรือ LOW (เปิด หรือปิด) ไปยังขา digital ที่กำหนดหมายเลขขาไอซีอาจกำหนดเป็นตัวแปรหรือค่าคงที่ (0-13) [3]

**2.2 ไมโครคอนโทรลเลอร์ (Microcontroller)**

ไมโครคอนโทรลเลอร์ (Microcontroller) คือ อุปกรณ์ควบคุมขนาดเล็ก ซึ่งบรรจุความสามารถที่คล้ายคลึงกับระบบคอมพิวเตอร์ โดยในไมโครคอนโทรลเลอร์ได้รวมเอาซีพียู, หน่วยความจำและพอร์ต ซึ่งเป็นส่วนประกอบหลักสำคัญของระบบคอมพิวเตอร์เข้าไว้ด้วยกัน โดยทำการบรรจุเข้าไว้ในตัวถังเดียวกันโครงสร้างโดยทั่วไปของไมโครคอนโทรลเลอร์นั้น สามารถแบ่งออกมาได้เป็น 5 ส่วนใหญ่ ๆ ดังต่อไปนี้

1. หน่วยประมวลผลกลางหรือซีพียู (CPU : Central Processing Unit)

2. หน่วยความจำ (Memory) สามารถแบ่งออกเป็น 2 ส่วน คือ หน่วยความจำที่มีไว้สำหรับเก็บโปรแกรมหลัก (Program Memory) เปรียบเสมือนฮาร์ดดิสก์ของเครื่องคอมพิวเตอร์ตั้งโต๊ะ คือข้อมูลใด ๆ ที่ถูกเก็บไว้ในนี้จะไม่สูญหายไปแม้ไม่มีไฟเลี้ยง อีกส่วนหนึ่งคือหน่วยความจำข้อมูล (Data Memory) ใช้เป็นเหมือนกับกระดาษทดในการคำนวณของซีพียู และเป็นที่พักข้อมูลชั่วคราวขณะทำงาน แต่หากไม่มีไฟเลี้ยง ข้อมูลก็จะหายไปคล้ายกับหน่วยความจำแรม (RAM) ในเครื่องคอมพิวเตอร์ทั่ว ๆ ไป แต่สำหรับไมโครคอนโทรลเลอร์สมัยใหม่ หน่วยความจำข้อมูลจะมีทั้งที่เป็นหน่วยความจำแรม ซึ่งข้อมูลจะหายไปเมื่อไม่มีไฟเลี้ยง และเป็นอีอีพรอม (EEPROM : Electrically Erasable Programmable Read-Only Memory) ซึ่งสามารถเก็บ ข้อมูลได้แม้ไม่มีไฟเลี้ยงก็ตาม

3. ส่วนติดต่อกับอุปกรณ์ภายนอก หรือพอร์ต (Port) มี 2 ลักษณะคือ พอร์ตอินพุต (Input Port) และพอร์ตส่งสัญญาณ หรือพอร์ตเอาต์พุต (Output Port) ส่วนนี้จะใช้ในการเชื่อมต่อกับอุปกรณ์ภายนอก ถือว่าเป็นส่วนที่สำคัญมาก ใช้ร่วมกันระหว่างพอร์ตอินพุต เพื่อรับสัญญาณ อาจจะด้วยการกดสวิตช์ เพื่อนำไปประมวลผล และส่งไปพอร์ตเอาต์พุต เพื่อแสดงผลเช่น การติดสว่างของหลอดไฟ เป็นต้น

4. ช่องทางเดินของสัญญาณ หรือบัส (BUS) คือเส้นทางการแลกเปลี่ยนสัญญาณข้อมูลระหว่าง ซีพียู หน่วยความจำ และพอร์ต เป็นลักษณะของสายสัญญาณจำนวนมากอยู่ภายในตัวไมโครคอนโทรลเลอร์ โดยแบ่งเป็นบัสข้อมูล (Data Bus), บัสแอดเดรส (Address Bus) และ บัสควบคุม (Control Bus)

5. วงจรกำเนิดสัญญาณนาฬิกา เป็นองค์ประกอบที่สำคัญมากอีกส่วนหนึ่ง เนื่องจากการทำงานที่เกิดขึ้นในตัวไมโครคอนโทรลเลอร์จะขึ้นอยู่กับการกำหนดจังหวะ หากสัญญาณนาฬิกามีความถี่สูง จังหวะการทำงานก็จะสามารถทำได้ถี่ขึ้นส่งผลให้ไมโครคอนโทรลเลอร์ตัวนั้น มีความเร็วในการประมวลผลสูงตามไปด้วยสำหรับไมโครคอนโทรลเลอร์ที่ผู้จัดทำเลือกใช้ คือ Arduino R3 ATMega 328p U แสดงดังรูปที่ 2 [4]

รูปภาพประกอบด้วย อุปกรณ์อิเล็กทรอนิกส์, วงจร

คำอธิบายที่สร้างโดยอัตโนมัติ

รูปที่ 2 คือ Arduino R3 ATMega 328p U

Arduino เป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษา ซึ่งส่วนประกอบของ Arduino R3 ATMega 328p Uแสดงดังรูปที่ 3 และรายละเอียดแสดดังตารางที่ 1 [5][6]

รูปภาพประกอบด้วย ภาพหน้าจอ

คำอธิบายที่สร้างโดยอัตโนมัติ

รูปที่ 3 ส่วนประกอบของ Arduino R3 ATMega 328p U

ตารางที่ 1 รายละเอียด Arduino R3 ATMega 328p U

|  |  |
| --- | --- |
| Microcontroller : | ATmega 328P (8 bit) |
| Operating Voltage : | 5 Volts |
| Digital I/O Pin : | 14 Pins |
| Analog Input Pin : | 6 Pins |
| DC Current per I/O Pin : | 20 mA |
| Flash Memory : | 32 KB |
| SRAM : | 2 KB |
| EEPROM : | 1 KB |
| Clock Speed : | 16 MHz |
| Size : | 53.4 x 68.6 mm |
| Weight : | 25 g |

**2.3 โมดูลเซ็นเซอร์แสงสำหรับตรวจจับวัตถุกีดขวาง IR Infrared Obstacle Avoidance Sensor Module**

โดยโมดูลนี้ มีส่วนประกอบได้ดังรูปที่ 4 โดยจะมีตัวรับและตัวส่ง infrared ในตัว ตัวสัญญาณ (สีขาว) infrared จะส่งสัญญาณออกมา และเมื่อมีวัตถุมาบัง คลื่นสัญญาณ infrared ที่ถูกสั่งออกมาจะสะท้อนกลับไปเข้าตัวรับสัญญาณ (สีดำ) สามารถนำมาใช้ตรวจจับวัตถุที่อยู่ตรงหน้าได้ และสามารถปรับความไว ระยะการตรวจจับ ใกล้หรือไกลได้สามารถแสดงได้ดังรูปที่ 5

รูปภาพประกอบด้วย ภาพหน้าจอ, แล็ปท็อป, จอภาพ, คอมพิวเตอร์

คำอธิบายที่สร้างโดยอัตโนมัติ

รูปที่ 4 ส่วนประกอบโมดูลเซ็นเซอร์แสงสำหรับตรวจจับวัตถุกีดขวาง

รูปภาพประกอบด้วย มือ, ถือ, โทรศัพท์

คำอธิบายที่สร้างโดยอัตโนมัติ

รูปที่ 5 การส่ง และรับสัญญาณ infrared

ภายตัวเซ็นเซอร์แบบนี้จะมีตัวส่ง Emitter และ ตัวรับ Receiver ติดตั้งภายในตัวเดียวกัน ทำให้ไม่จำเป็นต้องเดินสายไฟทั้งสองฝั่ง เหมือนแบบ Opposed Mode ทำให้การติดตั้งใช้งานได้ง่ายกว่า แต่อย่างไรก็ตามจำเป็นต้องติดตั้งตัวแผ่นสะท้อนหรือ Reflector ไว้ตรงข้ามกับตัวเซ็นเซอร์เอง โดยโฟโต้เซ็นเซอร์แบบที่ใช้แผ่นสะท้อนแบบนี้จะเหมาะสำหรับชิ้นงานที่มีลักษณะทึบแสงไม่เป็นมันวาว เนื่องจากอาจทำให้ตัวเซ็นเซอร์เข้าใจผิดว่าเป็นตัวแผ่นสะท้อน และ ทำให้ทำงานผิดพลาดได้ เซ็นเซอร์แบบนี้จะมีช่วงในการทำงาน หรือ ระยะในการตรวจจับจะได้ใกล้กว่าแบบ Opposed mode ซึ่งในสภาวะการทำงานปกติตัวรับ Receiver จะสามารถรับสัญญาณแสงจากตัวส่ง Emitter ได้ตลอดเวลา เนื่องจากลำแสงจะสะท้อนกับแผ่นสะท้อน Reflector อยู่ตลอดเวลา จะแสดงค่า เป็น 0 หน้าที่หลักของเซ็นเซอร์ชนิดนี้ จะคอยตรวจจับวัตถุที่เคลื่อนที่ตัดผ่านหน้าเซ็นเซอร์ เมื่อวัตถุ หรือ ชิ้นงานผ่านเข้ามาที่หน้าเซ็นเซอร์ แล้วจะการขวางลำแสงที่ส่งจากตัวส่ง Emitter ที่ส่งไปยังแผ่นสะท้อน จึงทำให้ตัวรับ Receiver ไม่สามารถรับลำแสงที่จะสะท้อนกลับมาได้ จะแสดงค่า เป็น 1 ซึ่งจะทำให้วงจรภายในรับรู้ได้ว่า มีวัตถุหรือชิ้นงานขวางอยู่ ทำให้สถานะของเอาต์พุตของตัวรับเปลี่ยนแปลงไป โดยเราเรียกลักษณะการทำงานแบบนี้ว่า Dark On หรือ Dark Operate ซึ่งคุณสมบัติของ IR Infrared Obstacle Avoidance Sensor Module สามารถแสดงได้ดังตารางที่ 2 [7]

ตารางที่ 2 คุณสมบัติของ IR Infrared Obstacle Avoidance Sensor Module

|  |  |
| --- | --- |
| ไฟเลี้ยง VCC : | 3.3-5V |
| ดิจิตอลเอาต์พุต (0 หรือ 1) | |
| ระยะตรวจจับ สามารถปรับได้ตั้งแต่ 2-30 cm | |
| มุมในการตรวจจับ 35 องศา | |
| ขนาดบอร์ด 3.1 x 1.5 cm | |

**2.4 เซ็นเซอร์วัดระยะทาง Ultrasonic Module**

เซนเซอร์วัดระยะทางด้วย Ultrasonic ใช้หลักการ ส่งคลื่นเสียงความถี่ต่ำ Ultrasonic ไปเมื่อคลื่นเสียงกระทบกับวัตถุจะมีการสะท้อนกลับมา เซนเซอร์จับเวลาที่ส่งคลื่นเสียงออกไปจนถึงคลื่นเสียงสะท้อนกลับมา เมื่อนำมาคำนวณกับเวลาที่เสียงเดินทางในอากาศ ก็จะได้ระยะทางออกมา

โมดูล Ultrasonic ตรวจจับวัตถุ คำนวณระยะทางโดยใช้คลื่น มีลักษณะเป็นกรวยและไม่ใช่เส้นตรง จึงเหมาะสำหรับใช้ตรวจจับสิ่งกีดขวางด้วย โดยมีส่วนประกอบโมดูล Ultrasonic ดังรูปที่ 6 และแสดงคุณสมบัติดังตารางที่ 3 [8]

รูปภาพประกอบด้วย ภาพหน้าจอ, จอภาพ, คอมพิวเตอร์

คำอธิบายที่สร้างโดยอัตโนมัติ

รูปที่ 6 ส่วนประกอบโมดูล Ultrasonic

ตารางที่ 3 คุณสมบัติเซ็นเซอร์วัดระยะทาง Ultrasonic Module

|  |  |
| --- | --- |
| ไฟเลี้ยง VCC : | 3.3-5V |
| ดิจิตอลเอาต์พุต (0 หรือ 1) | |
| ดิจิตอลอินพุต (0 หรือ 1) | |
| HC-SR04 : | ระยะตรวจจับ 2-400 cm |
| US-025 : | ระยะตรวจจับ 2-600 cm |

**2.5 DC Motor Speed Control**

ประกอบด้วย 2 ส่วนใหญ่ ๆ คือ 1) H-bridge Driver และ 2) Pulse-width modulation (PWM) สิ่งที่เป็นพลังขับเคลื่อนหลักให้กับหุ่นยนต์นั้น ก็คงจะไม่พ้น มอเตอร์ ซึ่งต้องการการควบคุม จาก Motor Driver ที่จะมาควบคุม ทั้ง ทิศทาง และ ความเร็ว ของมอเตอร์ซึ่งทางผู้จัดทำได้เลือกใช้ L298N Dual H-Bridge Motor Controller ซึ่งสามารถแสดงได้ดังรูปที่ 7 และคุณสมบัติของ L298N Dual   
H-Bridge Motor Controller ดังตารางที่ 4 โดยหลักการทำงานวงจร H-Bridge ของ L298N จะขับกระแสเข้ามอเตอร์ ตามขั้วที่กำหนดด้วยลอจิกเพื่อควบคุมทิศทาง ส่วนความเร็วของมอเตอร์นั้นจะถูกควบคุมด้วย สัญญาณ PWM (Pulse Width Modulation) เป็นวิธีการควบคุมการจ่ายกำลังโดยการปรับความกว้างของสัญญาณ Pulse ด้วยความถี่สูงเพื่อให้ได้กำลังเฉลี่ยเป็นไปตามส่วนที่ต้องการ ซึ่งต้องมีการปรับความถี่ให้เหมาะสมกับเป็นพารามิเตอร์ที่ใช้กำหนดสัดส่วนการทำงาน (ON) ของ Load (มอเตอร์) [9][10]

รูปภาพประกอบด้วย ภาพหน้าจอ, จอภาพ, คอมพิวเตอร์, นั่ง

คำอธิบายที่สร้างโดยอัตโนมัติ

รูปที่ 7 ส่วนประกอบของ L298N Dual H-Bridge Motor Controller

ตารางที่ 4 คุณสมบัติ L298N Dual H-Bridge Motor Controller

|  |  |
| --- | --- |
| Supply voltage : | 2-10V |
| Signal input voltage: : | 1.8-7V |
| Max output current : | 3A (1.5A\*2) |
| Control signal: | PWM |

**2.6 TCRT5000 Infrared Reflective sensor**

เป็นโมดูลตรวจจับวัตถุระยะใกล้ มีราคาถูก ขนาดเล็กสะดวกในการนำไปใช้ติดตั้งกับงานจำพวกหุ่นยนต์, Smart car, หุ่นยนต์หลบสิ่งกีดขวาง เป็นต้น ซึ่งสามารถแสดงได้ดังรูปที่ 8 และคุณสมบัติดังตารางที่ 5 โดยการทำงานของตัวโมดูลนี้ เริ่มต้นโดยให้ หลอด Infrared LED ทำการส่งสัญญาณเป็นแสงอินฟราเรดออกไปตกกระทบกับวัตถุที่ตรวจพบในระยะ และทำการสะท้อนกลับมายังตัวหลอด  
โฟโต้ไดโอดที่ทำหน้าที่รับแสงอินฟราเรด โดยส่วนมาก ตัวโมดูลจะให้ค่า output ออกมาเป็น Digital signal แต่สำหรับบางโมดูลอาจจะรองรับ output แบบ Analog signal ด้วย ส่วนตัว R ปรับค่านั้นใช้ในการปรับความไวต่อการตรวจจับแสงอินฟราเรด ซึ่งจะส่งผลต่อระยะในการตรวจพบวัตถุของตัวเซนเซอร์

รูปภาพประกอบด้วย วงจร

คำอธิบายที่สร้างโดยอัตโนมัติ

รูปที่ 8 ส่วนประกอบของ TCRT5000 Infrared Reflective sensor

ตารางที่ 4 คุณสมบัติเซ็นเซอร์วัดระยะทาง TCRT5000 Infrared Reflective sensor

|  |  |
| --- | --- |
| ไฟเลี้ยง VCC : | 3.3-5V |
| ดิจิตอลเอาต์พุต (0 หรือ 1) | |
| อนาล็อกเอาต์พุต | |

**2.7 วงจร DC/DC Step-up (แรงดันปรับค่าได้)**

เป็นวงจรที่ทำหน้าที่เพิ่มแรงดันไฟฟ้า โดยที่ทางผู้จัดทำเลือกใช้เป็นรุ่น MT3608 สามารถแสดงส่วนประกอบได้ดังภาพที่ 9 และสามารถและคุณสมบัติดังตารางที่ 6 [11]

รูปภาพประกอบด้วย วงจร

คำอธิบายที่สร้างโดยอัตโนมัติ

รูปที่ 9 ส่วนประกอบของ DC/DC Step-up รุ่น MT3608

ตารางที่ 6 คุณสมบัติของ DC/DC Step-up รุ่น MT3608

|  |  |
| --- | --- |
| กระแสไฟขาออกสูงสุด(I max) : | 2 A |
| แรงดันไฟฟ้าเข้า (V in): | 2 V ~ 24 V |
| แรงดันขาออกสูงสุด(V 0ut max) | 28 V |
| ประสิทธิภาพ (%Eff) | > 93% |

โดยกฎของโอห์มกระแสไฟฟ้านั่นวงจรไฟฟ้านั้น จะแปรผัน ตรงกับ แรงดันของแหล่งจ่ายไฟฟ้าแต่จะแปรผกผันกับค่า ความต้านทานในวงจรไฟฟ้า” ดังสมการ [12]

(1)

ข้อควรระวังเมื่อแรงดันเพิ่ม กระแสไฟไฟฟ้าจะต้องลดลงดังตัวอย่างที่แสดงในรูปที่ 10

รูปภาพประกอบด้วย นาฬิกา

คำอธิบายที่สร้างโดยอัตโนมัติ

รูปที่ 10 ตัวอย่างความสัมพันธ์ระหว่างแรงดันเพิ่มกับกระแสไฟลดลง

**2.8 ภาษา C#**

ภาษา C# (C-Sharp) เป็นภาษาโปรแกรมคอมพิวเตอร์ระดับสูงที่ใช้สำหรับเขียนโปรแกรมคอมพิวเตอร์ ซึ่งได้รับความนิยมเพิ่มมากขึ้นเรื่อย ๆ ในปัจจุบัน และเป็นภาษาโปรแกรมคอมพิวเตอร์ที่เหมาะสำหรับผู้ที่เริ่มต้นสนใจที่จะเขียนโปรแกรมคอมพิวเตอร์เป็นอย่างยิ่ง ซึ่งภาษา C# ถูกพัฒนามาจากภาษา C++ และมีโครงสร้างแบบเชิงวัตถุ (object-oriented programming) โดยใช้ Visual Studio เป็นเครื่องมือสำหรับพัฒนาโปรแกรมคอมพิวเตอร์ ซึ่ง Visual Studio เป็นเครื่องมือที่คอยอำนวยความสะดวกในการพัฒนาโปรแกรมคอมพิวเตอร์ ทำให้ผู้เขียนโปรแกรมสามารถพัฒนาโปรแกรมคอมพิวเตอร์ได้ไม่ยากนัก

ภาษา C# ได้รวบรวมข้อดีของภาษาต่าง ๆ เข้าไว้ด้วยกัน ไม่ว่าจะเป็นภาษา Java ภาษา C และ ภาษา C++ โดยมีข้อดีดังนี้

1) เป็นภาษาที่เขียนง่าย ไม่ซับซ้อนและเรียบง่าย เพราะคล้ายภาษา Java ภาษา C และ ภาษา C++ ทำให้หลายคนเข้าใจได้ไม่ยาก

2) เป็นภาษาโปรแกรมคอมพิวเตอร์ยุคใหม่ที่ถูกสร้างขึ้นมาสำหรับการพัฒนาโปรแกรมคอมพิวเตอร์ ภายใต้แนวคิด .NET Framework ซึ่งเป็นแนวคิดที่ได้รับความนิยมสูงที่สุดในปัจจุบัน

3) เป็นภาษาที่ถูกออกแบบมาให้ทำงานบน .NET Framework โดย .NET Framework เป็นรูปแบบในการพัฒนาโปรแกรมคอมพิวเตอร์สมัยใหม่ ซึ่งบริษัทไมโครซอพท์เป็นผู้พัฒนา คุณสมบัติที่สำคัญของ .NET Framework ก็คือ ผู้ใช้งานสามารถใช้งานบนระบบฮาร์ดแวร์ (Hardware) หรือ ระบบปฏิบัติการ (Operating System) ที่แตกต่างกันได้อย่างไม่มีปัญหา ดังนั้นผู้เขียนโปรแกรมจึงสามารถเขียนโปรแกรมคอมพิวเตอร์ใหม่ๆ ได้โดยง่าย รวดเร็ว และไม่ต้องติดข้อจำกัดต่าง ๆ อย่างเช่นการเขียนโปรแกรมคอมพิวเตอร์ในสมัยก่อนอีกต่อไป

4) เป็นภาษาที่แข็งแกร่งเพราะเป็นภาษาที่ได้มีการแก้ไขข้อบกพร่องบางอย่างของภาษา Java ภาษา C และ ภาษา C++ เหล่านั้น ทำให้ ภาษา C# เป็นภาษาที่มีความสมบูรณ์ตามแบบฉบับของโครงสร้างแบบเชิงวัตถุ (object-oriented programming) [13] ซึ่งโครงสร้างคำสั่งของโปรแกรมภาษา C# แสดงในรูปที่ 11

รูปภาพประกอบด้วย ภาพหน้าจอ

คำอธิบายที่สร้างโดยอัตโนมัติ

รูปที่ 11 โครงสร้างคำสั่งของโปรแกรมภาษา C#

Method ในภาษา C# เมธอดนั้นเป็นส่วนหนึ่งของโปรแกรมเพื่อให้ดำเนินการบางอย่าง กล่าวอีกนัยหนึ่ง คือ ส่วนย่อยของโปรแกรมที่ถูกเรียกโดยโปรแกรมหลัก และสามารถนำกลับมาใช้ซ้ำได้ การสร้างเมธอดจะช่วยให้ลดจำนวนของซอสโค้ดที่เขียนขึ้น ช่วยการนำกลับมาใช้ใหม่ และทำให้ง่ายในการแก้ไขโปรแกรมในภายหลัง แต่ เมธอด ในอีกคำหนึ่งที่รู้จัก คือ ฟังก์ชัน แต่คำว่าเมธอดใช้สำหรับฟังก์ชันที่อยู่ในคลาส (การเขียนโปรแกรมเชิงวัตถุ) [14] องค์ประกอบพื้นฐานที่สำคัญตามรูปแบบของ C# ที่สำคัญมีดังนี้

1) Block { ... }

ใช้รูปแบบของภาษาเช่นเดียวกับภาษา C/C++ คือ จะใช้บล็อก { ... } ในการกำหนดจุดเริ่มต้น และจุดสิ้นสุดของการทำงานในแต่ละส่วน ซึ่งภายในบล็อกอาจมีบล็อกย่อย ๆ ซ้อนลงไปได้อีกตามลักษณะของงาน เช่น

if (……………..)

{

for (……………..)

{

……….

}

……….

}

ทั้งนี้เครื่องหมาย “ { ” (open brace) กับ “ } ’’ (close brace) นั้นต้องครบคู่กันพอดี และโดยทั่วไปเรานิยมวาง “ { ” ไว้ให้ตรงกับคำสั่งเริ่มต้นบล็อก เช่น จากรหัสคำสั่งตัวอย่างที่เราวางไว้ตรงกับคำสั่ง if หรือ for เป็นต้น ทั้งนี้เพื่อให้พิจารณาขอบเขตของบล็อกได้ง่าย แต่ไม่ใช่กฎข้อบังคับแต่อย่างใด สามารถเลือกวางในแบบที่ถนัดก็ได้

2) เครื่องหมายสิ้นสุดคำสั่ง ( ; )

ใน C# จะใช้เครื่องหมายเซมิโคลอน (semicolon) “ ; ’’ เป็นตัวแสดงจุดสิ้นสุดของแต่ละคำสั่ง หากไม่ใส่เครื่องหมายนี้เพื่อคั่นระหว่างคำสั่ง โปรแกรมจะถือว่าเป็นคำสั่งเดียวกันไปตลอดแม้ว่าจะอยู่คนละบรรทัดก็ตาม เช่น จากตัวอย่างมีคำสั่งทั้งหมด 3 คำสั่ง

x = 10;

y = “xxx”;

z = x + 10;

เมื่อเครื่องหมาย ; เป็นตัวบ่งบอกจุดสิ้นสุดของคำสั่ง จึงสามารถมีคำสั่งอยู่ในบรรทัดเดียวกันมากกว่า 1 คำสั่งก็ได้ เช่น จากตัวอย่างมีคำสั่งทั้งหมด 3 คำสั่ง

x = 10; y = “xxx”; z = x + 10;

การเขียนรหัสคำสั่งในลักษณะนี้ ไม่ค่อยนิยมทำกัน เนื่องจากจะทำให้เราโปรแกรมได้ยาก รหัสโปรแกรมดูไม่เป็นระเบียบ แต่ก็อาจนำมาใช้ในบางกรณีได้เช่นกัน

บางคำสั่งไม่สามารถทำให้จบภายในบรรทัดเดียวได้ เพราะบางคำสั่งจะมีคำสั่งย่อย ๆ อยู่ภายในอีกก็ได้ ดังนั้นคำสั่งเหล่านี้จะมีบล็อกของตนเอง เช่น คำสั่ง if, คำสั่ง for และ คำสั่ง while เป็นต้น เมื่อคำสั่งใดมีบล็อกอยู่แล้วก็ไม่จำเป็นต้องใส่เครื่องหมาย ; ต่อท้ายเครื่องหมายบล็อกอีก แต่คำสั่งภายในต้องมีเครื่องหมาย ; ตามปกติ เช่น

if (เงื่อนไข)

{

คำสั่ง A;

คำสั่ง B;

}

3) การเขียนคำอธิบายประกอบแทรกไว้ในรหัสโปรแกรม

คำอธิบาย (Comment) หมายถึง การเขียนข้อความใดๆ ที่ไม่ใช่คำสั่งปะปนกันไปกับ คำสั่งอื่นๆ เพื่ออธิบายเรื่องใดเรื่องหนึ่งเอาไว้เพื่อความเข้าใจในรหัสโปรแกรมตรงส่วนนั้น ดังนั้นเพื่อให้โปรแกรมไม่สับสนว่าส่วนใดเป็นคำสั่ง ส่วนใดเป็นเพียงคำอธิบายไม่เกี่ยวข้องกับการทำงานของโปรแกรม จึงต้องใช้เครื่องหมายมาเป็นตัวช่วยในการแยกแยะ การแทรกคำอธิบายสามารถทำได้ 2 ลักษณะดังนี้ คือ

รูปแบบ // คำอธิบาย

ใช้สำหรับคำอธิบายแบบบรรทัดเดียว (Single Line comment) โดยโปรแกรมจะถือว่าตั้งแต่สัญลักษณ์ // เป็นต้นไปจนถึงสิ้นสุดบรรทัดเป็นคำอธิบายทั้งหมด จะไม่นำมาพิจารณาในการประมวลผล เช่น

// สูตรคำนวณหาขนาดพื้นที่สี่เหลี่ยมผืนผ้า

RectangleArea=width\*length;

หรือ

RectangleArea=width\*length; // สูตรคำนวณหาขนาดพื้นที่สี่เหลี่ยมผืนผ้า

รูปแบบ /\* คำอธิบาย \*/

กรณีที่คำอธิบายของเราค่อนข้างยาวจำเป็นต้องเขียนหลายๆ บรรทัด การใช้ // หลายๆ ครั้งอาจไม่สะดวกนัก เราก็สามารถใช้ /\*...\*/ ครอบคำอธิบายนั้นแทน โดยโปรแกรมจะถือว่าตั้งแต่สัญลักษณ์ /\* เป็นต้นไปจะเป็นคำอธิบายทั้งหมดจนกว่าจะเจอสัญลักษณ์ \*/ จึงจะถือว่าเป็นการสิ้นสุดคำอธิบาย เช่น /\* นี่คือส่วนของคำอธิบายที่ไม่มีผลต่อการทำงานของโปรแกรมมีไว้เพื่อช่วยให้ทำความเข้าใจรหัสคำสั่งได้ง่ายขึ้น \*/ [15]