



โครงสร้าง และการออกแบบโปรแกรม

สำหรับโปรแกรมที่ผู้วิจัยได้พัฒนาเป็นโปรแกรมคอมพิวเตอร์ โดยถูกพัฒนาขึ้นมาด้วยโปรแกรม Microsoft Visual Studio .NET 2008 บนระบบปฏิบัติการ Microsoft Windows XP โดยเน้นที่การใช้งานที่ง่าย สะดวก เข้าใจได้ง่ายตามลักษณะของโปรแกรมทั่วไปในระบบปฏิบัติการนี้ และในการเลือกใช้โปรแกรม Microsoft Visual Studio .NET 2008 ในการพัฒนาโปรแกรม เนื่องจากโปรแกรม Microsoft Visual Studio .NET 2008 เป็นโปรแกรมที่

- มีการทำงานในลักษณะของการ โปรแกรมเชิงวัตถุ (Object Oriented Programming) ทำให้สะดวกในการเขียนโปรแกรม และทำความเข้าใจกับ source code
- ใช้พัฒนาโปรแกรมบนระบบปฏิบัติการ Windows ซึ่งเป็นระบบปฏิบัติการที่ใช้งานง่าย และแพร่หลายกว่าระบบปฏิบัติการอื่น
- มีเครื่องมือต่างๆ ในการพัฒนาโปรแกรมให้ใช้เป็นจำนวนมาก ทำให้สามารถใช้เวลาส่วนใหญ่นกับการพัฒนาส่วนที่เป็นแกนหลักของโปรแกรม
- สามารถสร้างไฟล์โปรแกรมได้เป็นไฟล์เดียว โดยไม่ต้องมีไฟล์อื่นประกอบ ทำให้ง่ายในการติดตั้งเพื่อให้ใช้งาน และลดความผิดพลาดจากการติดตั้งไฟล์ไม่ครบด้วย
- มีระบบช่วยเหลือที่ดี และเพียงพอต่อการเรียนรู้เพื่อพัฒนาโปรแกรม

การทำงานของโปรแกรมเป็นลักษณะตัวแปลภาษา (Interpreter) โดยจะอ่านชุดรหัสคำสั่งทีละบรรทัด ทำการเก็บค่าคำสั่ง และค่าตัวแปรของคำสั่ง แล้วนำค่าเหล่านั้นไปใช้ในการควบคุมการเคลื่อนที่ของแกนกลแบบขนานในตระกูล H-4 กับโต๊ะหมุน รวมเป็น 5 แกน ในการออกแบบผู้วิจัยได้แบ่งส่วนประกอบของโปรแกรมออกเป็น 2 ส่วน ดังนี้

1. ส่วนการแปลเอ็นซีโค้ด ในส่วนนี้เป็นส่วนหนึ่งที่มีความสำคัญ เพราะเป็นส่วนที่ทำหน้าที่อ่านข้อมูลจีโค้ด จากแฟ้มข้อมูลแบบ Text แล้วทำการแปลงให้อยู่ในรูปแบบรหัสคำสั่งที่พร้อมนำไปใช้ควบคุม
2. ส่วนติดต่อกับผู้ใช้ (User Interface) เป็นส่วนที่ผู้ใช้ติดต่อกับโปรแกรมคอมพิวเตอร์ที่ผู้วิจัยสร้างขึ้นผ่านทางหน้าจอโปรแกรมคอมพิวเตอร์

3.1 การออกแบบส่วนการแปลเอ็นซีโค้ด

จากรูปแบบคำสั่งของจีโค้ด ที่ได้จากเพิ่มข้อมูลแบบ Text สามารถจำแนกกลุ่มข้อมูล ออกเป็น 2 กลุ่มคือ กลุ่มที่เป็นชุดคำสั่ง (Command) และกลุ่มที่เป็นพารามิเตอร์ (Parameter) ที่ตามหลังชุดคำสั่ง ทั้งกลุ่มที่เป็นชุดคำสั่ง และกลุ่มที่เป็นพารามิเตอร์จะประกอบไปด้วยตัวอักษร (Key) แล้วตามด้วยค่าที่เป็นตัวเลข (Value) โดยตัวอักษรจะแสดงชนิดของชุดคำสั่งหรือชนิดของพารามิเตอร์ ส่วนค่าที่เป็นตัวเลขจะเป็นตัวกำหนดค่าของชุดคำสั่งที่ใช้ หรือเป็นค่าของพารามิเตอร์

ในการแปลงคำสั่งจีโค้ด จากเพิ่มข้อมูลแบบ Text จะทำการอ่านชุดของตัวอักษรเข้ามาเพื่อแปลงทีละบรรทัด จากนั้นจะเปรียบเทียบตัวอักษรทีละตัวในบรรทัดนั้นว่าเป็น Command key, Command value, Parameter key หรือ Parameter value โดยที่

1. Command key จะเป็นกลุ่มของตัวอักษร G, M และ N
2. Parameter key จะเป็นกลุ่มตัวอักษร X, Y, Z, F, I, J, K, P, R, S และ T
3. Command value จะเป็นตัวเลขที่ตามหลัง Command key
4. Parameter value ก็จะเป็นตัวเลขที่ตามหลัง Parameter key

ดังนั้นเมื่อแปลงข้อมูลจากเพิ่มข้อมูลแบบ Text แล้วจะได้เป็น 2 กลุ่ม คือ Command object ซึ่งประกอบไปด้วย Command key, Command value และกลุ่มของ Parameter object ซึ่ง Parameter object แต่ละตัวก็จะประกอบไปด้วย Parameter key และ Parameter value เมื่อ Parser ทำการแปลงข้อมูลที่อ่านเข้ามาในแต่ละบรรทัดเสร็จแล้ว จะส่ง Object ที่ได้ทั้งหมดจากการแปลงในบรรทัดนั้นไปเก็บไว้ แล้วจึงทำการแปลงในบรรทัดต่อไป

จากแนวคิดข้างต้นเราจะนำมาสร้างเป็น โปรแกรมโดยใช้ภาษา C++ โดยจะสร้างเป็นคลาสขึ้นมา 3 คลาส (Class) ได้แก่ Parser class, Command class และ Parameter class โดยแต่ละ คลาส จะมีหน้าที่แตกต่างกันไป

Parser class จะทำหน้าที่รับชื่อของเพิ่มข้อมูลแบบ Text ที่ต้องการจะแปลง แล้วอ่านข้อมูลเข้ามาทีละบรรทัด จากนั้นแปลงข้อมูลที่อ่านเข้ามาให้อยู่ในรูปของชุดคำสั่ง (Command object) แล้วทำการจับคู่ (Map) คำสั่ง กับฟังก์ชันการทำงานของระบบ

ในส่วนของ Command class และ Parameter class จะทำหน้าที่ในการเก็บข้อมูลของ Command object และ Parameter object ตามลำดับ

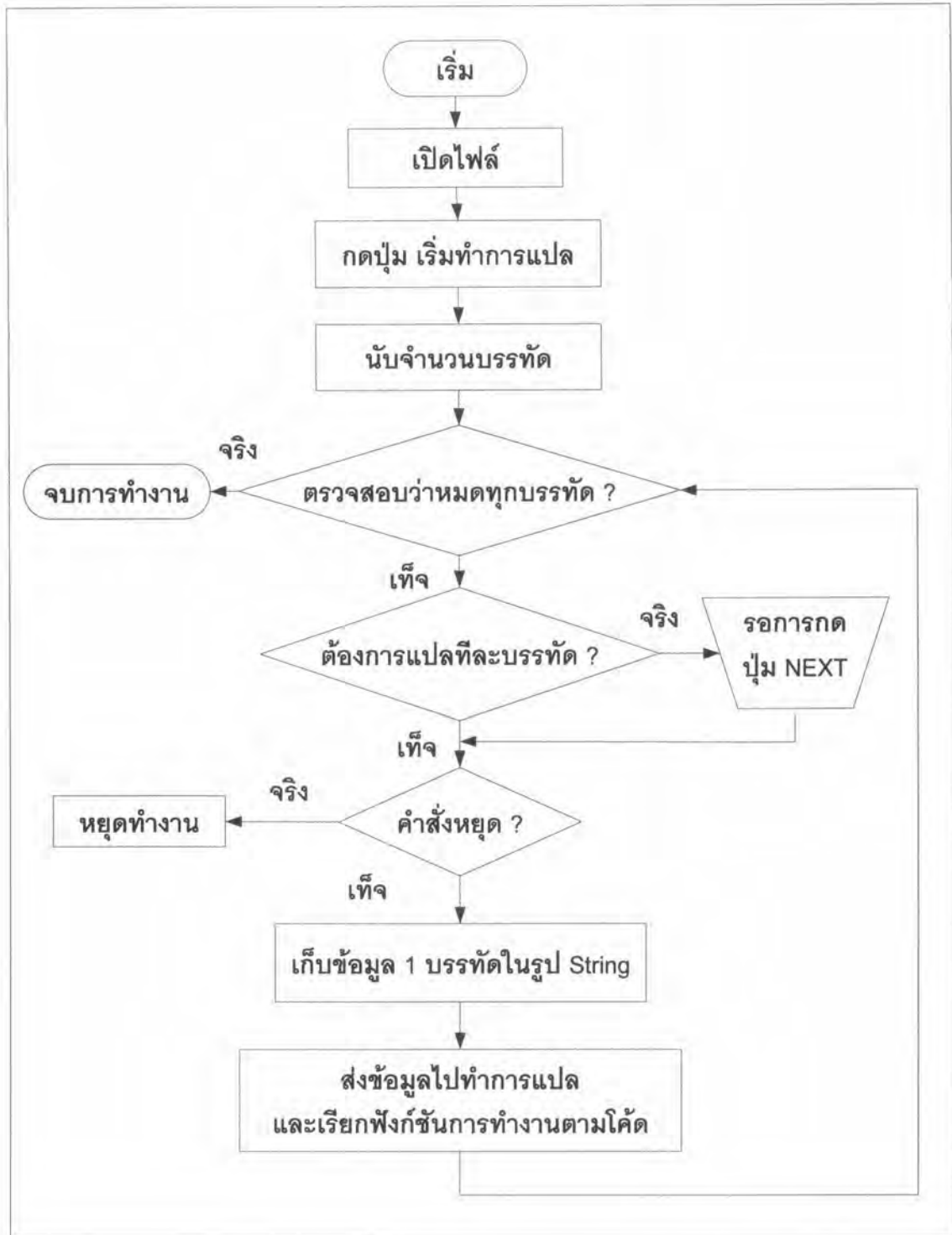
3.1.1 การรับข้อมูลเข้าสู่โปรแกรม

การรับข้อมูลเริ่มจากการเปิดไฟล์ชนิดเท็กไฟล์ (*.txt) เข้ามาในโปรแกรม เมื่อมีการกดปุ่ม Step หรือกดปุ่ม Continuous ซึ่งเป็นปุ่มคำสั่งให้เริ่มการแปล โปรแกรมจะทำการนับว่าในไฟล์ข้อมูลมีอยู่ที่บรรทัด เพื่อนำค่าจำนวนบรรทัดไปใช้ในการอ้างอิงว่าทำการแปลข้อมูลเสร็จสิ้นทุกบรรทัดหรือยัง

จากนั้นจะเข้าสู่คู่มือการทำงาน โดยเริ่มจากการตรวจสอบว่าตำแหน่งของบรรทัดนี้ มีค่าน้อยกว่าจำนวนบรรทัดทั้งหมดที่มีอยู่ในไฟล์หรือไม่ หากค่าน้อยกว่าแสดงว่ายังไม่ใช่บรรทัดสุดท้ายของข้อมูลให้ทำการแปลต่อไป

ขั้นต่อมาจะมีการตรวจสอบว่าผู้ใช้งานต้องการวิธีการแปลเป็นแบบใด โดยแบบแรก หากผู้ใช้กดปุ่ม Step จะเป็นการแปลทีละบรรทัดแล้วหยุดรอคำสั่ง คือเมื่อมีการแปลเสร็จเรียบร้อยแล้ว โปรแกรมจะรอจนกว่าผู้ใช้จะกดปุ่ม Next จึงจะทำการแปลบรรทัดต่อไป แบบที่สอง หากผู้ใช้กดปุ่ม Continuous จะเป็นการแปลอย่างต่อเนื่องทีละบรรทัด คือเมื่อมีการแปลในหนึ่งบรรทัดเสร็จเรียบร้อยแล้ว จะทำการแปลบรรทัดต่อไปทันที

จากนั้นเป็นการตรวจสอบว่ามีการกดปุ่ม Stop ให้หยุดการทำงานหรือไม่ หลังจากนั้นจะทำการเก็บข้อมูลบรรทัดนั้นในรูป String แล้วส่งข้อมูลไปทำการแปล รอจนกระทั่งส่วนที่ทำการแปลส่งสัญญาณว่าทำงานเสร็จแล้วจึงวนกลับไปทำซ้ำคู่มือการทำงานใหม่จนกระทั่งบรรทัดสุดท้ายจึงจบการทำงาน ดังรูปที่ 3.1 แผนภูมิโครงสร้างการรับข้อมูล



รูปที่ 3.1 แผนภูมิโครงสร้างการรับข้อมูลเข้าสู่โปรแกรม

3.1.2 การแปลข้อมูล

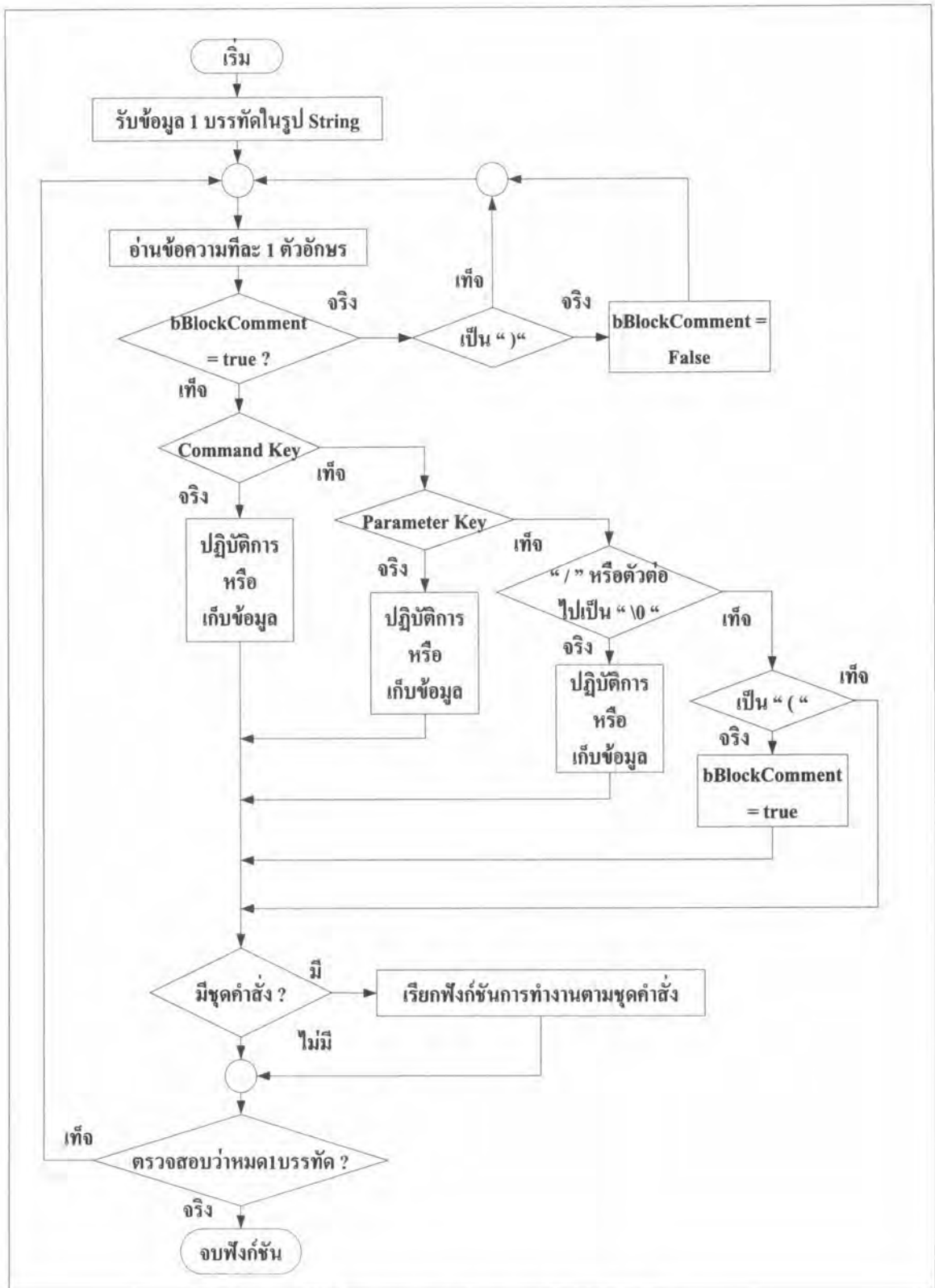
เมื่อรับข้อความมาทีละ 1 บรรทัดแล้ว จะทำการรับค่าตัวอักษรทีละ 1 ตัวอักษร จากนั้นจะทำการตรวจสอบว่าตัวอักษรนั้นอยู่ในเครื่องหมายวงเล็บหรือไม่ โดยเครื่องหมายวงเล็บจะเป็นส่วนที่แสดงว่าจะไม่มีการแปลโค้ดในส่วนนั้น (Block comment) หากอยู่ในส่วนที่อยู่ในเครื่องหมายวงเล็บค่าของตัวแปร `bBlockComment` จะเป็น true ก็จะมีการตรวจสอบต่อไปว่าตัวอักษรนั้นเป็นเครื่องหมายวงเล็บปิด “) ” หรือไม่ หากยังก็จะกลับไปรับตัวอักษรตัวใหม่ แล้วกลับมาตรวจสอบอีกจนกว่าจะออกจากส่วนที่อยู่ในเครื่องหมายวงเล็บ

ในส่วนต่อมา จะตรวจสอบว่าตัวอักษรนั้นอยู่ในกลุ่มใด ใน 4 กลุ่ม ดังนี้

1. Command key จะเป็นกลุ่มของตัวอักษร G, M และ N
2. Parameter key จะเป็นกลุ่มตัวอักษร X, Y, Z, F, I, J, K, P, R, S และ T
3. เครื่องหมาย “ / ” (Block skip) หรือเครื่องหมาย “ \0 ” (NULL)
4. เครื่องหมายวงเล็บเปิด

ในกลุ่มที่ 1 Command key จะเป็นตัวอักษรที่แสดงถึงชนิดของคำสั่ง กลุ่มที่ 2 Parameter key เป็นตัวอักษรที่แสดงถึงตัวแปรที่เป็นรายละเอียดของคำสั่ง กลุ่มที่ 3 เครื่องหมาย “ / ” (Block skip) แสดงว่าจะไม่มีการแปลในบรรทัดนั้น และเครื่องหมาย “ \0 ” (NULL) ซึ่งเป็นตัวสุดท้ายของข้อมูล 1 บรรทัด แสดงถึงการหมดบรรทัด กลุ่มที่ 4 เครื่องหมายวงเล็บเปิด แสดงถึงการจุดเริ่มต้นส่วนที่จะไม่มีการแปลโค้ด (Block comment) โดยในแต่ละกลุ่มจะมีการปฏิบัติการ และการเก็บค่าของข้อมูลแตกต่างกันออกไป

จากนั้นจะตรวจสอบว่า มีชุดคำสั่งอยู่หรือไม่ หากมีการสร้างชุดคำสั่งแล้ว ก็จะเรียกไปยังฟังก์ชันการทำงานเพื่อทำงานตามชุดคำสั่งนั้น เมื่อทำงานตามชุดคำสั่งเสร็จแล้ว ก็จะกลับไปตรวจสอบว่าข้อมูลหมดบรรทัดหรือไม่ แต่ถ้าไม่มีการสร้างชุดคำสั่งไว้ จะทำการตรวจสอบว่าข้อมูลหมดบรรทัดหรือไม่ทันที หากหมดบรรทัดแล้ว ก็จะจบฟังก์ชัน และกลับไปรับข้อมูลบรรทัดใหม่เข้ามาทำการแปล หากยังไม่หมดบรรทัดก็จะกลับไปรับตัวอักษรตัวถัดมา มาทำการแปลต่อไป ดังรูปที่ 3.2



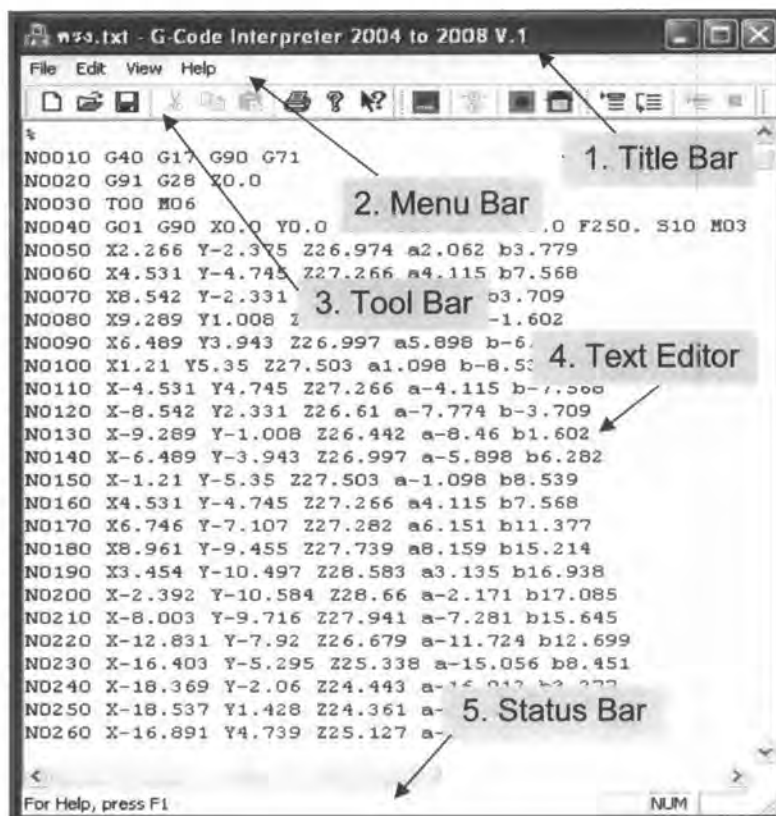
รูปที่ 3.2 แผนภูมิโครงสร้างการแปลซีโค้ด

3.2 การออกแบบส่วนติดต่อกับผู้ใช้ (User Interface)

ในส่วนนี้จะแนะนำส่วนของหน้าจอโปรแกรม คำสั่งและปุ่มการทำงานต่าง ในส่วนของวิธีการใช้โปรแกรมจะอธิบายรายละเอียดในส่วนภาคผนวก ข

หน้าจอโปรแกรมแบ่งออกเป็น 5 ส่วนคือ

1. ส่วนแสดงชื่อโปรแกรมและชื่อไฟล์ที่กำลังทำงาน (Title Bar)
2. แถบเมนูควบคุมการทำงานของโปรแกรม (Menu Bar)
3. แถบปุ่มควบคุมการทำงานของโปรแกรม (Tool Bar)
4. หน้าต่างเพื่อแสดงและทำการแก้ไขข้อมูลจีโค้ด (Text Editor)
5. แถบแสดงสถานะต่างๆ (Status Bar)



รูปที่ 3.3 ส่วนประกอบของหน้าจอโปรแกรม

3.2.1 ส่วนแสดงชื่อโปรแกรมและชื่อไฟล์ที่กำลังทำงาน (Title Bar)

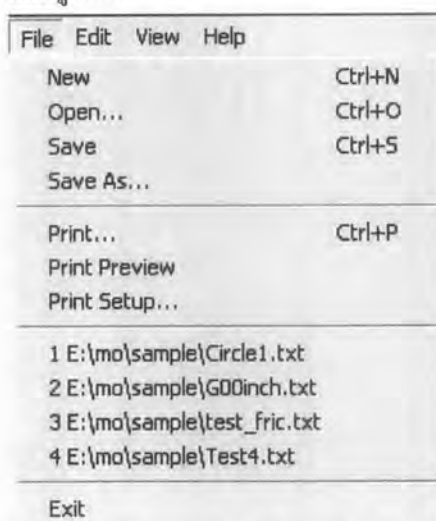
เป็นส่วนที่แสดงชื่อไฟล์ที่กำลังทำงาน และชื่อ โปรแกรม โดยชื่อไฟล์ที่กำลังทำงานอยู่ตามด้วยชื่อโปรแกรม ซึ่งขึ้นด้วยเครื่องหมาย “ - ”



3.2.2 แถบเมนูควบคุมการทำงานของโปรแกรม (Menu Bar)

ประกอบด้วยเมนูหลักดังรูป File Edit View Help โดยเมนู File ทำหน้าที่เกี่ยวกับการจัดการเกี่ยวกับไฟล์ข้อมูล Edit จัดการเกี่ยวกับการแก้ไขข้อความ View จัดการเกี่ยวกับการแสดงแถบทูล และไดอะล็อก และ Help ระบบช่วยเหลือ

- รายละเอียดในเมนูไฟล์



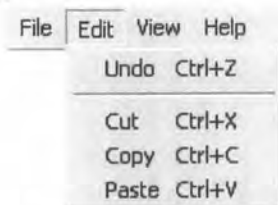
รูปที่ 3.4 รายละเอียดในเมนูไฟล์

เมนู	การทำงาน
New	สร้างไฟล์ G-Code ใหม่
Open	เปิดไฟล์ G-Code เดิม
Save	บันทึกการเปลี่ยนแปลงไฟล์ G-Code
Save as	บันทึกการเปลี่ยนแปลงไฟล์ G-Code เป็นชื่อใหม่
Print	พิมพ์ไฟล์ G-Code ออกทางเครื่องพิมพ์
Print Preview	แสดงตัวอย่างการพิมพ์
Print Setup	ตั้งค่าเครื่องพิมพ์

Exit

ออกจากโปรแกรม

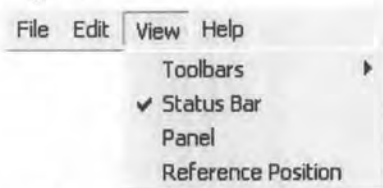
- รายละเอียดในเมนู Edit



รูปที่ 3.5 รายละเอียดในเมนู Edit

เมนู	การทำงาน
Undo	ยกเลิกการแก้ไขครั้งสุดท้าย
Cut	ตัดข้อความที่เลือกออก
Copy	คัดลอกข้อความที่เลือก
paste	วางข้อความที่คัดลอก

- รายละเอียดในเมนู View

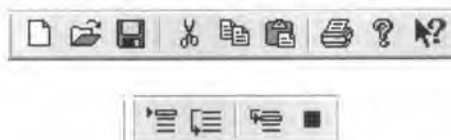


รูปที่ 3.6 รายละเอียดในเมนู View

เมนู	การทำงาน
Toolbars	เปิด/ปิดทูลบาร์ที่ต้องการ
Status bar	เปิด/ปิดสเตตัสบาร์
Panel	เปิด/ปิดจอแสดงผล
Referance Position	เปิด/ปิดไอคอนการชดเชย โคออร์ดิเนตของชิ้นงาน

3.2.3 แถบปุ่มควบคุมการทำงานของโปรแกรม (Tool Bar)

เป็นแถบปุ่มสำหรับควบคุมการทำงานของโปรแกรม โดยสร้างปุ่มสำหรับการทำงานที่เกิดขึ้นบ่อย เพื่อให้การทำงานสามารถกระทำได้สะดวก ซึ่งสามารถแสดงได้ดังรูปที่ 3.7



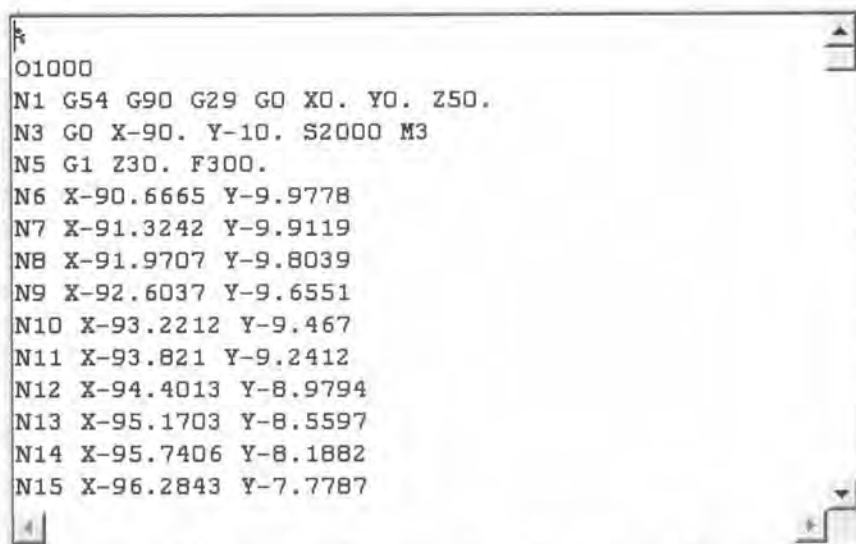
รูปที่ 3.7 หน้าจอโปรแกรมส่วนแถบปุ่ม

ปุ่ม	การทำงาน
(New)	สร้างไฟล์ G-Code ใหม่
(Open)	เปิดไฟล์ G-Code เดิม
(Save)	บันทึกการเปลี่ยนแปลงไฟล์ G-Code
(Cut)	ตัดข้อความที่เลือกออก
(Copy)	คัดลอกข้อความที่เลือก
(Paste)	วางข้อความที่คัดลอก
(Print)	พิมพ์ไฟล์ G-Code ออกทางเครื่องพิมพ์
(Help)	ระบบช่วยเหลือ

ปุ่ม	การทำงาน
(Step)	แปลจีโค้ดทีละบรรทัด แล้วหยุดรอการกดปุ่มคำสั่ง Next
(Continuous)	แปลจีโค้ดทีละบรรทัด ทำอย่างต่อเนื่อง ไม่หยุดรอคำสั่ง Next
(Next)	แปลจีโค้ดทีละบรรทัดต่อไป แล้วหยุดรอการกดปุ่มคำสั่ง Next
(Stop)	หยุดแปลจีโค้ด

3.2.4 หน้าต่างเพื่อแสดงและทำการแก้ไขข้อมูลจีโค้ด (Text Editor)

เป็นส่วนที่แสดงจีโค้ด คล้ายกับการใช้โปรแกรม Notepad สามารถทำการแก้ไขโค้ด พิมพ์หรือลบโค้ดที่ไม่ต้องการได้ สามารถใช้คำสั่งคัดลอก ตัด วางข้อความ ในการแก้ไขโค้ด



```

O1000
N1 G54 G90 G29 GO XO. YO. Z50.
N3 GO X-90. Y-10. S2000 M3
N5 G1 Z30. F300.
N6 X-90.6665 Y-9.9778
N7 X-91.3242 Y-9.9119
N8 X-91.9707 Y-9.8039
N9 X-92.6037 Y-9.6551
N10 X-93.2212 Y-9.467
N11 X-93.821 Y-9.2412
N12 X-94.4013 Y-8.9794
N13 X-95.1703 Y-8.5597
N14 X-95.7406 Y-8.1882
N15 X-96.2843 Y-7.7787
  
```

รูปที่ 3.8 หน้าจอ โปรแกรมส่วนหน้าต่างแสดงโค้ด

3.2.5 แถบแสดงสถานะต่างๆ (Status Bar)

แถบแสดงสถานะซึ่งอยู่ด้านล่างของหน้าต่างโปรแกรม จะแสดงข้อความอธิบายสั้นๆ เมื่อนำเมาส์ (mouse) ชี้อะบบปุ่ม หรือแถบปุ่ม



รูปที่ 3.9 หน้าจอ โปรแกรมส่วนแถบแสดงสถานะ