บทที่ 3

วิธีการดำเนินการวิจัย

3.1 การกำหนดความสามารถของโปรแกรม

จากการศึกษาโปรแกรมการจำลองการทำงานของเครื่องกลึงซีเอ็นซี 3 โปรแกรมคือ

- CNC Basic Software version 5.17 ของ R. & S. Keller GmbH
- CNC Simulator Turning version 1.0
- Novaturn PC ของ Denford Machine Tools Ltd.

พบข้อจำกัดต่างๆ ในแต่ละโปรแกรมคือ

- เป็นโปรแกรมที่ไม่ใช่ภาษาไทย ทำให้เป็นเรื่องยุ่งยากของผู้ใช้ที่เป็นคนไทย
- เป็นโปรแกรมที่ทำงานบนระบบปฏิบัติการดอส(DOS)ซึ่งการติดต่อระหว่างผู้ใช้กับโปรแกรมยังทำได้ไม่สะดวกนัก
- ไม่สามารถทำการชดเชยค่าจมูกมีดได้โดยอัตโนมัติ ทั้งที่การชดเชยค่าจมูกมีดเป็นสิ่งสำคัญที่จะทำให้ได้ขนาดของชิ้นงานที่มีรูปร่างและขนาดถูกต้องแม่นยำ
- มีข้อจำกัดในการเขียนรหัสสั่งการทำงานมากกว่าที่สามารถเขียนได้จริงในเครื่องกลึงซีเอ็นซีทั่วไป เช่นไม่สามารถเขียนคำสั่งสลับก่อนหลังสำหรับค่า X และ Z ได้ หรือไม่สามารถสั่งคำสั่ง G ได้มากกว่า 1 คำสั่งในแต่ละบรรทัดเป็นต้น
- ไม่สามารถพิมพ์ออกทางเครื่องพิมพ์ได้โดยง่าย
- ผู้ใช้ไม่สามารถทำการเปลี่ยนแปลงขนาด หรือเพิ่มชนิดของมีดตัดได้เองโดยง่าย
- ผู้ใช้ไม่สามารถกำหนดมีดตัด และค่าออฟเซตของมีดตัดแต่ละตัวได้โดยง่าย
- ไม่มีระบบช่วยเหลือ สำหรับผู้ที่เรียนรู้ด้วยตัวเอง

สำหรับโปรแกรมที่พัฒนาขึ้นนี้เป็นการพัฒนาขึ้นเพื่อลดข้อจำกัดต่างๆข้างต้นลง โปรแกรมพัฒนาขึ้นด้วยโปรแกรม Borland C++ Builder 4 บนระบบปฏิบัติการ Microsoft Windows 98 ที่ความละเอียดของจอภาพขนาด 1024x768จุด โดยเน้นที่การใช้งานที่ง่าย สะดวก และเข้าใจได้ง่ายตามลักษณะของโปรแกรมทั่วไปในระบบปฏิบัติการนี้ และในการเลือกใช้โปรแกรม Borland C++ Builder 4 ในการพัฒนาโปรแกรม เนื่องจาก Borland C++ Builder 4 เป็นโปรแกรมที่

- รองรับการแสดงผลเป็นภาษาไทยบนระบบปฏิบัติการที่รองรับการใช้งานภาษาไทย

- ใช้พัฒนาโปรแกรมบนระบบปฏิบัติการ Windows ซึ่งเป็นระบบปฏิบัติการที่ใช้งานง่ายและแพร่หลายกว่าระบบปฏิบัติการอื่นๆ
- มีการทำงานในลักษณะของการโปรแกรมเชิงวัตถุ(object oriented programming) ทำให้สะดวกต่อการเขียนโปรแกรม และทำความเข้าใจกับ source code
- มีเครื่องมือต่างๆในการพัฒนาโปรแกรมให้ใช้เป็นจำนวนมาก ทำให้สามารถใช้เวลาส่วนใหญ่ไปกับการพัฒนาส่วนที่เป็นแกนหลักของโปรแกรม โดยไม่ต้องเสียเวลาไปกับการพัฒนาโปรแกรมในทุกๆรายละเอียด ทำให้ระยะเวลาในการพัฒนาโปรแกรมสั้นลง
- สามารถสร้างไฟล์โปรแกรมได้เป็นไฟล์เดียวโดยไม่จำเป็นต้องมีไฟล์อื่นๆประกอบ ทำให้ง่ายในการติดตั้งเพื่อใช้งาน และลดความผิดพลาดจากการติดตั้งไฟล์ไม่ครบด้วย
- มีระบบช่วยเหลือที่ดี และเพียงพอต่อการเรียนรู้เพื่อพัฒนาโปรแกรม

การทำงานของโปรแกรมเป็นในลักษณะการทำงานแบบตัวแปลภาษา(Interpreter) คือแสดงผลของชุดรหัสสั่งการทำงานทีละบรรทัดจนครบ โดยก่อนที่จะเริ่มจำลองการทำงานนั้นโปรแกรมจะทำการตรวจสอบรูปแบบการเขียนชุดรหัสสั่งการทำงานถ้าหากไม่มีข้อผิดพลาดจึงจะเริ่มทำการจำลองการทำงาน โปรแกรมแบ่งการทำงานเป็น 3 ส่วนหลัก คือ

- ส่วนเขียนชุดรหัสสั่งการทำงาน เป็นส่วนที่ทำหน้าที่ในการเขียน การแก้ไข การจัดเก็บ การนำกลับมาใช้ใหม่ การพิมพ์ชุดรหัสสั่งการทำงานออกทางเครื่องพิมพ์ และการระบุรายละเอียดของงาน ตลอดจนขนาดของชิ้นงาน และการระบุมีดตัดและค่าชดเชยต่างๆของหัวตัดอันได้แก่ ค่าออฟเซต และค่าชดเชยจมูกมีด
- ส่วนตรวจสอบรูปแบบ และความถูกต้องของชุดรหัสสั่งการทำงาน
- ส่วนจำลองการทำงาน ที่สามารถจำลองการทำงานทีละบรรทัด หรือต่อเนื่องก็ได้ โดยแสดงผลของชุดรหัสสั่งการทำงานในรูปของภาพเคลื่อนไหว

สำหรับการเปลี่ยนแปลงหรือเพิ่มรูปแบบของหัวตัดนั้นผู้ใช้โปรแกรมสามารถที่จะใช้โปรแกรมวาดภาพทั่วไปสร้างรูปหัวตัดตามที่ต้องการแล้วบันทึกเป็นไฟล์รูปภาพ(bitmap file)ที่มีนามสกุล .bmp ได้ ก็สามารถที่จะนำไฟล์นั้นมาใช้งานร่วมกับโปรแกรมได้ทันที โดยโปรแกรมจะบันทึกข้อมูลของการระบุมีดตัด ไว้ในไฟล์ข้อความ(text file)ชื่อ tools.cfg และข้อมูลของออฟเซตไว้ในไฟล์ข้อความชื่อ offset.cfg ซึ่งไฟล์ทั้งสองจะบันทึกไว้ในโฟลเดอร์(folder) เดียวกับที่ไฟล์โปรแกรม และไฟล์รูปภาพของมีดตัดอยู่

3.2 หลักการการทำงานของโปรแกรม

3.2.1 โปรแกรมประกอบด้วยไฟล์ต่างๆ ที่เกี่ยวข้อง ดังนี้

- ไฟล์โปรแกรม (MainMenu.EXE) เพื่อควบคุมการทำงานทั้งหมดของโปรแกรม
- ไฟล์ข้อมูลชุดรหัสสั่งการทำงาน (Text File) เก็บข้อมูลรายละเอียดของชิ้นงานและชุดรหัสสั่งการทำงาน
- ไฟล์ข้อมูลมีดตัด (Tools.CFG) เก็บข้อมูลของการระบุเลือกใช้มีดตัด เพื่อการอ้างอิงถึงในชุดรหัสสั่งการทำงาน
- ไฟล์ข้อมูลค่าออฟเซต (Offset.CFG) เก็บข้อมูลของการระบุค่าออฟเซต และการชดเชยจมูกมีด เพื่อการอ้างอิงถึงในชุดรหัสสั่งการทำงาน
- ไฟล์รูปภาพมีดตัด (Bitmap File)
- ไฟล์ช่วยเหลือ (RookIE.HLP)

3.2.2 รูปแบบของไฟล์ข้อมูลชุดรหัสสั่งการทำงาน จะต้องเป็นไฟล์ข้อความ(Text File)โดยข้อมูลในแต่ละบรรทัดจะเป็นดังนี้

- บรรทัดที่ 1 สำหรับเก็บข้อมูลชื่องาน
- บรรทัดที่ 2 สำหรับเก็บข้อมูลชื่อชิ้นงาน
- บรรทัดที่ 3 สำหรับเก็บข้อมูลชนิดวัสดุ
- บรรทัดที่ 4 สำหรับเก็บข้อมูลขนาดเส้นผ่านศูนย์กลางของชิ้นงาน ที่ตำแหน่งแรกของบรรทัดจะเป็นตัวเลขแสดงหน่วยซึ่ง 0 แทนหน่วยนิ้ว และ 1 แทนหน่วย ม.ม. และตามด้วยขนาดของชิ้นงานโดยไม่มีการเว้นวรรค
- บรรทัดที่ 5 สำหรับเก็บข้อมูลขนาดความยาวของชิ้นงาน ที่ตำแหน่งแรกของบรรทัดจะเป็นตัวเลขแสดงหน่วยซึ่ง 0 แทนหน่วยนิ้ว และ 1 แทนหน่วย ม.ม. และตามด้วยขนาดของชิ้นงานโดยไม่มีการเว้นวรรค
- บรรทัดที่ 6 เก็บข้อมูลชื่อผู้เขียนโปรแกรม
- บรรทัดที่ 7 เป็นต้นไป เก็บข้อมูลชุดรหัสสั่งการทำงาน

3.2.3 รูปแบบของไฟล์ข้อมูลมีดตัด จะต้องเป็นไฟล์ข้อความ(Text File)โดยข้อมูลในแต่ละบรรทัดจะเป็นดังนี้

- บรรทัดที่ 1 เก็บชื่อไฟล์รูปภาพสำหรับมีดตัดหมายเลข 01
- บรรทัดที่ 2 เก็บชื่อไฟล์รูปภาพสำหรับมีดตัดหมายเลข 02

- บรรทัดที่ 3 เก็บชื่อไฟล์รูปภาพสำหรับมีดตัดหมายเลข 03
- บรรทัดที่ 4 เก็บชื่อไฟล์รูปภาพสำหรับมีดตัดหมายเลข 04
- บรรทัดที่ 5 เก็บชื่อไฟล์รูปภาพสำหรับมีดตัดหมายเลข 05
- บรรทัดที่ 6 เก็บชื่อไฟล์รูปภาพสำหรับมีดตัดหมายเลข 06
- บรรทัดที่ 7 เก็บชื่อไฟล์รูปภาพสำหรับมีดตัดหมายเลข 07
- บรรทัดที่ 8 เก็บชื่อไฟล์รูปภาพสำหรับมีดตัดหมายเลข 08
- บรรทัดอื่นถัดจากนี้ไปไม่ได้ใช้

3.2.4 รูปแบบของไฟล์ข้อมูลค่าออฟเซต จะต้องเป็นไฟล์ข้อความ(Text File)โดยข้อมูลในแต่ละบรรทัดจะเป็นดังนี้

- บรรทัดที่ 1 – 16 เก็บข้อมูลสำหรับค่าออฟเซตหมายเลข 1 – 16 ตามลำดับ โดยที่ในแต่ละบรรทัดจะประกอบด้วยตัวเลข 4 ชุดคือ ออฟเซตแกน X ออฟเซตแกน Y รัศมีจมูกมีด และตัวเลขปลายมีด โดยตัวเลขแต่ละตัวจะแยกกันไว้ด้วยช่องว่าง 1 ช่อง
- บรรทัดอื่นถัดจากนี้ไปไม่ได้ใช้

3.2.5 รูปแบบของไฟล์รูปภาพมีดตัด เป็น bitmap File ใช้อัตราส่วน 2 จุดต่อ 1 ม.ม. และมีข้อกำหนดของการใช้สีเป็น 3 ส่วนคือ

- ส่วนที่เป็นสีดำ (0x000000) แทนอุปกรณ์ยึดจับมีดตัด
- ส่วนที่เป็นสีขาว (0xFFFFFF) แทนพื้นที่ว่าง
- สีอื่นๆ แทนส่วนที่เป็นคมตัดของมีดตัดแต่ละตัว

3.2.6 การทำงานของโปรแกรมเมื่อสั่งเริ่มต้นการจำลองการทำงานของเครื่องกลึงซีเอ็นซี จะเริ่มจากอ่านข้อมูลจากส่วนข้อมูลรหัสสั่งการทำงาน มาทีละบรรทัดและทำการตรวจสอบไวยากรณ์แล้วทำการกำหนดค่าเริ่มต้นต่างๆ วาดรูปชิ้นงาน แล้วจึงเริ่มทำการจำลองการทำงานไปทีละบรรทัด โดยถ้ามีความผิดพลาดของชุดรหัสสั่งการทำงานเมื่อไร โปรแกรมจะหยุดการจำลองการทำงานทันที

3.2.7 ส่วนตรวจสอบไวยากรณ์ เป็นการตรวจสอบความผิดพลาดของชุดรหัสสั่งการทำงาน โดยพิจารณาจากรูปแบบของการเขียน โดยมีข้อกำหนด หรือกฎเกณฑ์ต่างๆดังนี้

- อักขระที่ใช้ในการเขียนชุดรหัสสั่งการทำงานประกอบด้วย C, F, G, H, I, K, M, N, O, S, T, U, W, X และ Z เท่านั้น
- ในแต่ละบรรทัดของรหัสสั่งการทำงานประกอบด้วยคำสั่งต่างๆ
- แต่ละคำสั่งจะต้องไม่เขียนติดกัน
- ในแต่ละคำสั่งจะต้องประกอบด้วยอักขระและค่าตัวเลขที่เขียนติดกันอยู่ด้านหลังอักขระ ไม่สามารถเขียนเฉพาะอักขระ หรือค่าตัวเลขอยู่แยกต่างหากได้
- ค่าตัวเลขสำหรับอักขระแต่ละตัวมีข้อกำหนดดังแสดงในตารางที่ 3.1
- อักขระ C, F, H, I, K, N, O, R, S, T, U, W, X, และ Z มีได้เพียง 1 ตัวอักษรในแต่ละบรรทัด
- M00 M01 และ M02 ต้องอยู่เป็นคำสั่งเดี่ยวในบรรทัด
- M03 M04 และ M05 ต้องไม่อยู่ในบรรทัดเดียวกัน
- M08 และ M09 ต้องไม่อยู่ในบรรทัดเดียวกัน
- M41 และ M91 ต้องไม่อยู่ในบรรทัดเดียวกัน
- M86 และ M87 ต้องไม่อยู่ในบรรทัดเดียวกัน
- ในแต่ละบรรทัดมีอักขระ G ที่อยู่ในกลุ่มเดียวกันไม่ได้
- ในแต่ละบรรทัดมีอักขระ G ที่อยู่ในกลุ่ม 00 พร้อมกับอักขระ G ในกลุ่ม 01 ไม่ได้
- G98 และ G99 ต้องอยู่เป็นคำสั่งเดี่ยวในบรรทัด
- อักขระ O มีได้เพียงที่เดียวคือคำแรกสุดของโปรแกรม และต้องเป็นคำเดียวในบรรทัดนั้นด้วย
- อักขระ N เป็นได้เฉพาะคำแรกของบรรทัด
- อักขระ C และ H อยู่ในบรรทัดเดียวกันไม่ได้
- อักขระ X และ U อยู่ในบรรทัดเดียวกันไม่ได้
- อักขระ Z และ W อยู่ในบรรทัดเดียวกันไม่ได้
- ในบรรทัดที่มีคำสั่ง G04 จะต้องไม่มีอักขระ U, W หรือ Z
- อักขระตัวใหญ่ (Capital Letter) หรืออักขระตัวเล็ก(Small Letter) ไม่แตกต่างกันในการเขียนชุดคำสั่ง
- ลำดับก่อนหลังของการเขียนคำสั่งในแต่ละบรรทัดไม่มีผลต่อการทำงาน
- ขนาดของชิ้นงานต้องมากกว่าศูนย์ทั้งขนาดเส้นผ่านศูนย์กลาง และความยาวของชิ้นงาน

ตารางที่ 3.1 แสดงรูปแบบค่าตัวเลขของอักขระแต่ละตัว

อักขระ	ค่าตัวเลข	ขอบเขต	
		ค่าต่ำสุด	ค่าสูงสุด
C	จำนวนจริง	-9999.999	9999.999
F	จำนวนจริงบวก	0.001	9999.999
G	00, 01, 02, 03, 04, 20, 21, 27, 28, 29, 32, 40, 41, 42, 50, 96, 97, 98 และ 99		
H	จำนวนจริง	-9999.999	9999.999
I	จำนวนจริง	-9999.999	9999.999
K	จำนวนจริง	-9999.999	9999.999
M	00, 01, 02, 03, 04, 05, 08, 09, 41, 86, 87 และ 91		
N	จำนวนเต็ม	1	9999
O	จำนวนเต็ม	1	9999
S	จำนวนจริงบวก	0.001	9999.999
T	ประกอบด้วยตัวเลข 4 หลัก 2หลักแรกคือ 01, 02, 03, 04, 05, 06, 07และ 08 2 หลักหลังคือ 00, 01, 02, 03, 04, 05, 06, 07, 08, 09,10,11,12,13,14,15 และ 16		
U	จำนวนจริง	-9999.999	9999.999
W	จำนวนจริง	-9999.999	9999.999
X	จำนวนจริง	-9999.999	9999.999
Z	จำนวนจริง	-9999.999	9999.999

3.2.8 ส่วนตรวจสอบความผิดพลาดระหว่างการทำงาน เป็นความผิดพลาดที่ไม่ได้เกิดจากการเขียนชุดรหัสสั่งการทำงานผิดรูปแบบ แต่เกิดจากลำดับขั้นตอนของการสั่งการทำงาน และข้อจำกัดต่างๆของเครื่องกลึงซีเอ็นซีแต่ละเครื่อง ได้แก่

- ความเร็วรอบสูงสุด
- การสั่งการเคลื่อนที่ในคำสั่ง G01, G02, G03 หรือ G32 จะต้องระบุค่าความเร็วของการป้อนตัดชิ้นงานด้วย
- ในการเคลื่อนที่ภายใต้คำสั่ง G00 จะต้องไม่เกิดการตัดชิ้นงาน
- ในการเคลื่อนที่ตัดชิ้นงานจะต้องเกิดขึ้นเมื่อมีการหมุน
- ถ้าเป็นการระบุค่าความเร็วของการป้อนตัดชิ้นงานในหน่วยของ มม./รอบ จะต้องระบุความเร็วรอบของชิ้นงานด้วย

- ถ้าเป็นการระบุค่าความเร็วของการป้อนตัดชิ้นงานในหน่วยของ ม.ม./รอบ จะสามารถสั่งให้มีดตัดเคลื่อนที่ภายใต้คำสั่ง G01, G02, G03 หรือ G04 ได้ก็ต่อเมื่อชิ้นงานกำลังหมุนอยู่เท่านั้น
- การเปลี่ยนทิศทางการหมุนของชิ้นงานจะต้องทำการสั่งหยุดการหมุนก่อน ถึงจะสั่งหมุนชิ้นงานในทิศทางตรงข้ามได้
- ในขณะที่เบรกทำงานอยู่ไม่สามารถสั่งหมุนชิ้นงานได้
- ในขณะที่ชิ้นงานถูกสั่งหมุนอยู่ไม่สามารถสั่งเบรกได้ ต้องสั่งหยุดหมุนชิ้นงานเสียก่อน
- การสั่งทำเกลียวจะต้องสั่งในขณะที่เครื่องทำงานในลักษณะของการควบคุมความเร็วรอบคงที่ ไม่ใช่ความเร็วผิวคงที่
- อักขระ C หรือ H ไม่สามารถสั่งได้เมื่อเครื่องซีเอ็นซีไม่ได้ทำงานในอยู่โหมดของการควบคุมการเคลื่อนที่ของแกน C
- การจะสั่งให้เครื่องซีเอ็นซีทำงานในโหมดของการควบคุมการเคลื่อนที่ของแกน C นั้นจำเป็นที่จะต้องสั่งเบรกให้ทำงานเสียก่อน
- ไม่สามารถยกเลิกการทำงานของเบรคได้เมื่อเครื่องซีเอ็นซีทำงานอยู่ในโหมดของการควบคุมการเคลื่อนที่ของแกน C

3.2.9 ส่วนกำหนดค่าเริ่มต้นสำหรับจำลองการทำงาน ประกอบด้วยค่าต่างๆ คือ

- การเคลื่อนที่เป็นการเคลื่อนที่ภายใต้คำสั่ง G00
- หน่วยที่ใช้ในการทำงานเป็นหน่วย มิลลิเมตร
- ไม่ทำการชดเชยค่าจมูกมีด
- ทำงานภายใต้โหมดการทำงานแบบความเร็วรอบในการหมุนชิ้นงานคงที่
- ระบุอัตราการป้อนตัดในหน่วยของ ม.ม./รอบ
- ชิ้นงานไม่หมุน
- เบรกไม่ทำงาน
- สารหล่อเย็นไม่ทำงาน
- ไม่ได้ทำงานอยู่ในโหมดของการควบคุมการเคลื่อนที่ของแกน C
- ตำแหน่งเริ่มต้นของมีดตัดเป็น X=0 Z=300.

3.2.10 การจำลองการทำงานในแต่ละบรรทัด

- ถ้าเป็นคำสั่ง M จะทำตามคำสั่งทันที
- ถ้าเป็นการระบุค่าตำแหน่ง (อักขระ C, H, I, K, U, W, X หรือ Z) จะกำหนดให้แก่ตัวแปรต่างๆทันที
- ถ้าเป็นคำสั่งที่เกี่ยวข้องกับการเคลื่อนที่ของมีดตัด (G00 ,G01 ,G02 ,G03 ,G04, G27, G28, G32 หรือ G50) จะทำตามคำสั่งเมื่ออ่านคำสั่งอื่นๆจนครบทั้งบรรทัด
- คำสั่ง G อื่นๆ จะทำตามคำสั่งทันที

3.2.11 แนวคิดในการจำลองการเคลื่อนที่ เพื่อให้ได้การเคลื่อนที่ที่ดูต่อเนื่องสมจริง การเคลื่อนที่ของรูปมีดตัดชิ้นงานในแต่ละครั้งจะต้องไม่เกิน 1 จุดของการแสดงผลบนจอคอมพิวเตอร์

3.2.12 หลักการจำลองการเคลื่อนที่ภายใต้คำสั่ง G00 เนื่องจากการเคลื่อนที่ภายใต้คำสั่ง G00 เป็นการเคลื่อนที่ไปด้วยความเร็วสูงสุดของการเคลื่อนที่ในแต่ละแกน โดยที่การเคลื่อนที่ในแนวแกน Z จะเร็วกว่าการเคลื่อนที่ในแนวแกน X อยู่ 2 เท่า ระยะเวลาสำหรับการเคลื่อนที่ 1 จุดจึงใช้ความเร็วของการเคลื่อนที่ในแนวแกน X มาใช้ในการคำนวณ และในการเคลื่อนที่ในแนวแกน X 1 ครั้ง จะมีการเคลื่อนที่ในแนวแกน Z 2 ครั้ง โดยสามารถอธิบายตามลำดับได้ดังนี้

- เมื่อเริ่มต้นโปรแกรม ทำการคำนวณระยะเวลาสำหรับการเคลื่อนที่ 1 จุดบนจอคอมพิวเตอร์ในแนวแกน X จากสมการ

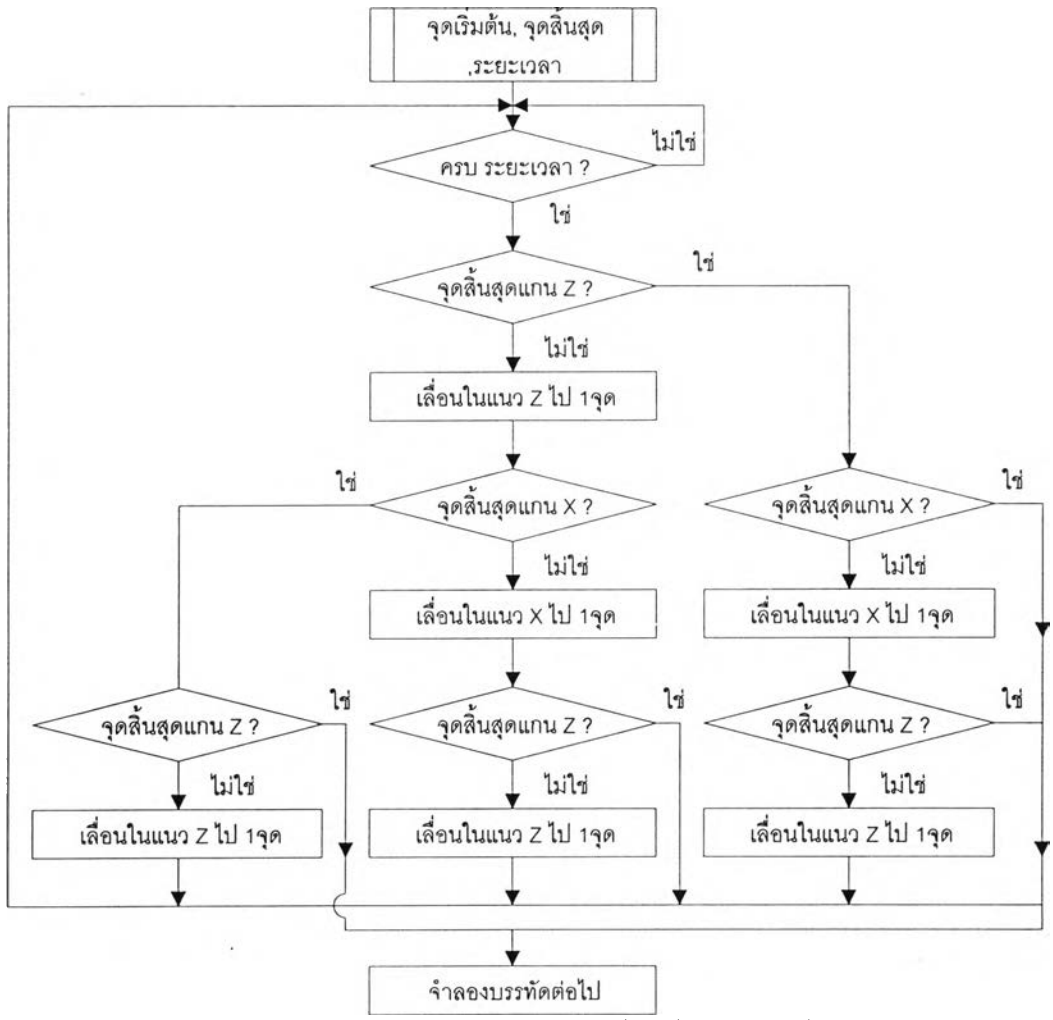
  ระยะเวลา = 60 x 1000 / (RapidFeed x DisplayScale)

  โดยที่ RapidFeed คือความเร็วสูงสุดของมีดตัดในการเคลื่อนที่ในแนวแกน X

  DisplayScale คือ คืออัตราส่วนระหว่างระยะทางบนจอคอมพิวเตอร์และระยะทางจริงบนเครื่องกลึงซีเอ็นซี หน่วยเป็น จุดต่อมิลลิเมตร
- เก็บค่าตำแหน่งปัจจุบันไว้เป็นค่าเริ่มต้น
- อ่านค่าตำแหน่งใหม่เพื่อเป็นค่าสำหรับสิ้นสุดการเคลื่อนที่
- ทุกๆ ระยะเวลาเท่ากับที่คำนวณไว้ ทำการตรวจสอบตามลำดับ
- ตรวจสอบตำแหน่งของมีดตัดในแนวแกน Z ว่าอยู่บนตำแหน่งสิ้นสุดหรือยัง ถ้ายังเคลื่อนที่ไปในทิศทางการเคลื่อนที่ในแนวแกน Z 1 จุด
- ตรวจสอบตำแหน่งของมีดตัดในแนวแกน X ว่าอยู่บนตำแหน่งสิ้นสุดหรือยัง ถ้ายังเคลื่อนที่ไปในทิศทางการเคลื่อนที่ในแนวแกน X 1 จุด

- ตรวจสอบตำแหน่งของมีดตัดในแนวแกน Z ว่าอยู่บนตำแหน่งสิ้นสุดหรือยัง ถ้ายังเคลื่อนที่ไปในทิศทางการเคลื่อนที่ในแนวแกน Z 1 จุด
- จะหยุดการเคลื่อนที่ และจำลองการทำงานในบรรทัดต่อไปเมื่อ มีดตัดมาอยู่ในตำแหน่งสิ้นสุดทั้ง 2 แกน

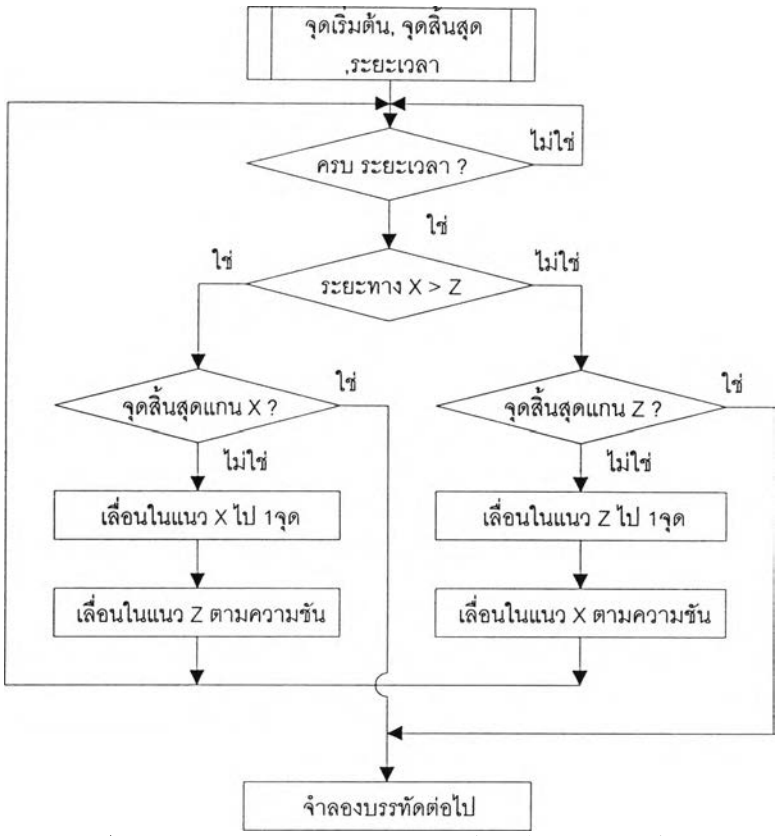
โดยสามารถแสดงด้วยแผนภูมิการไหลดังรูปที่ 3.2



รูปที่ 3.2 แผนภูมิการไหลของการเคลื่อนที่ภายใต้คำสั่ง G00

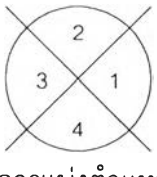
3.2.13 **หลักการจำลองการเคลื่อนที่ภายใต้คำสั่ง** G01 **เนื่องจากการเคลื่อนที่ภายใต้คำสั่ง** G01 **เป็นการเคลื่อนที่เป็นเส้นตรงด้วยความเร็วตามอัตราการป้อนตัด เพื่อให้การเคลื่อนที่เป็นไปอย่างต่อเนื่องและราบเรียบ จึงจะให้การเคลื่อนที่ในแนวแกนที่มีระยะทางมากกว่าเคลื่อนที่ไปทีละ 1 จุดบนจอคอมพิวเตอร์ และคำนวณการเคลื่อนที่ในอีกแนวแกนหนึ่งตามค่าความชันของแนวการเคลื่อนที่ โดยสามารถแสดงได้ดังแผนภูมิการไหลดังรูปที่ 3.3**

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

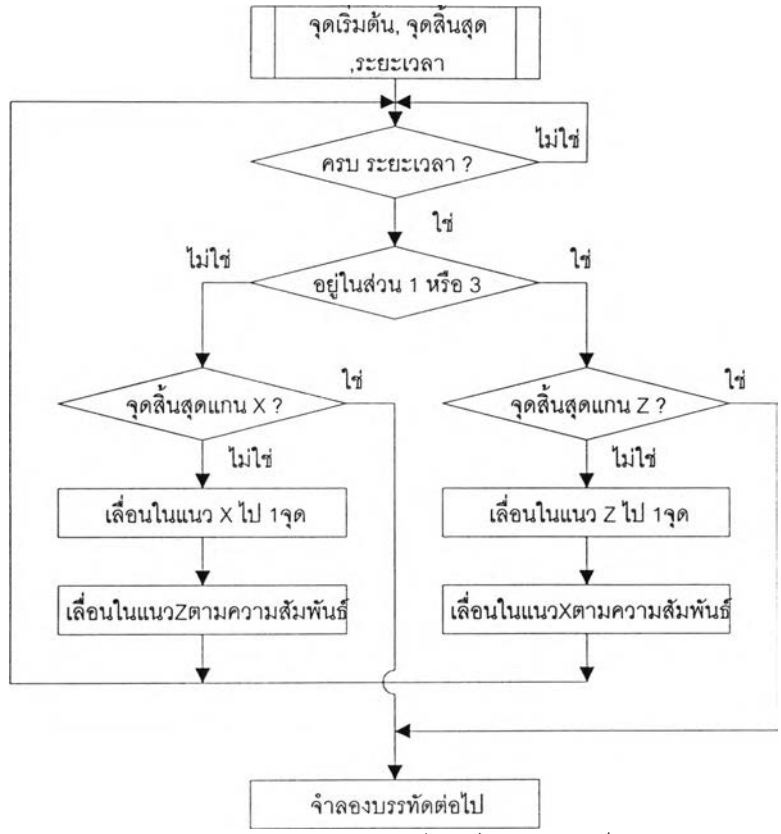


รูปที่ 3.3 แผนภูมิการไหลของการเคลื่อนที่ภายใต้คำสั่ง G01

3.2.14 **หลักการจำลองการเคลื่อนที่ภายใต้คำสั่ง** G02 และ G03 เนื่องจากการเคลื่อนที่ภ
คำสั่ง G02 หรือ G03 เป็นการเคลื่อนที่ในแนวโค้งด้วยความเร็วตามอัตราการป้อนตัด เพื่อให้
เคลื่อนที่เป็นไปอย่างต่อเนื่องและราบเรียบ จะแบ่งตำแหน่งการเคลื่อนที่เป็น 4 ส่วน ดังรูปที่ 3.4
การเคลื่อนที่ที่อยู่ในส่วนที่ 1 และ 3 ระยะทางในการเคลื่อนที่ในแนวแกน Z จะมากกว่าระยะทาง
เคลื่อนที่ในแนวแกน X จึงให้การเคลื่อนที่ในแนวแกน Z เป็นหลักคือเคลื่อนที่ไปทีละ 1 จุด บ
คอมพิวเตอร์ แล้วจึงคำนวณระยะทางการเคลื่อนที่ในแนวแกน X ตามความสัมพันธ์ของวงกลม และ
ทำนองเดียวกันเมื่อเคลื่อนที่ในส่วนที่ 2 และ 4 ซึ่งระยะทางในแนวแกน X มากกว่าระยะทางใน
แกน Z ก็จะให้การเคลื่อนที่ในแนวแกน X เป็นหลักคือเคลื่อนที่ไปทีละ 1 จุด บนจอคอมพิวเตอร์ แล้ว
คำนวณระยะทางการเคลื่อนที่ในแนวแกน Z ตามความสัมพันธ์ของวงกลม โดยสามารถแสดงได้
แผนภูมิการไหลดังรูปที่ 3.5



รูปที่ 3.4 แสดงการแบ่งตำแหน่งการเคลื่อนที่



รูปที่ 3.5 แผนภูมิการไหลของการเคลื่อนที่ภายใต้คำสั่ง G02 หรือ G03

3.2.15 หลักการจำลองการเคลื่อนที่ภายใต้คำสั่ง G27 และ G28 เนื่องจากเป็นการเคลื่อนที่ด้วยความเร็วสูงสุดของแต่ละแกนจึงใช้หลักการเดียวกับการจำลองการเคลื่อนที่ภายใต้คำสั่ง G00

3.2.16 การคำนวณระยะเวลาระหว่างการเคลื่อนที่แต่ละครั้งตามอัตราการป้อนตัด แบ่งได้เป็น 2 กรณีคือ

- อัตราการป้อนตัดเป็น ม.ม./รอบ ได้ระยะเวลา เท่ากับ

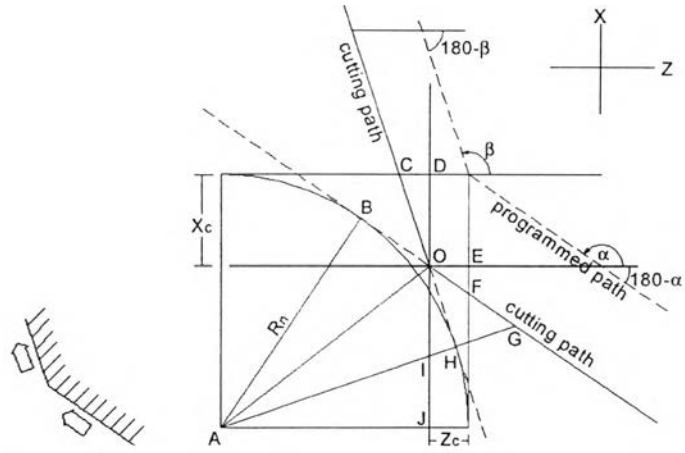
  60*1000/(FeedRate*SpindleSpeed*DisplayScale);

- อัตราการป้อนตัดเป็น ม.ม./นาที ได้ระยะเวลา เท่ากับ

  60*1000/(FeedRate*DisplayScale);

เมื่อ DisplayScale คืออัตราส่วนระหว่างระยะทางบนจอคอมพิวเตอร์และระยะทางจริงบนเครื่องกลึงซีเอ็นซี หน่วยเป็น จุดต่อมิลลิเมตร

3.2.17 การชดเชยจมูกมีด สามารถคำนวณได้ดังนี้



รูปที่ 3.6 แสดงการชดเชยค่าจมูกมีดในการควบคุมการเคลื่อนที่ของมีดตัดที่ Tool Tip =1 ใน G41

จากรูปที่ 3.6 ได้ค่าของมุมต่างๆ ดังนี้

$$JAI = HOI = EOI - EOH$$

$$= 90 - (180 - \beta)$$

$$= \beta - 90$$

$$BAG = GOH = HOE - GOE$$

$$= (180 - \beta) - (180 - \alpha)$$

$$= \alpha - \beta$$

$$HAO = BAO = BAG / 2$$

$$= (\alpha - \beta) / 2$$

$$JAO = JAI + HAO$$

$$= \beta - 90 + (\alpha - \beta) / 2$$

$$= ((\alpha + \beta) / 2) - 90$$

$$Zc = Rn - AJ$$

$$= Rn - AO \cos (JAO)$$

$$= Rn - \frac{Rn}{\cos(BAO)} . \cos(JAO)$$

$$Zc = Rn \left[1 - \frac{\sin\left(\frac{\alpha+\beta}{2}\right)}{\cos\left(\frac{\alpha-\beta}{2}\right)} \right]$$

และในทำนองเดียวกัน Xc = $Rn.\left[1+\frac{\cos\left(\frac{\alpha+\beta}{2}\right)}{\cos\left(\frac{\alpha-\beta}{2}\right)}\right]$

โดยการคำนวณในลักษณะเดียวกันสรุปสมการสำหรับการชดเชยค่าจมูกมีดได้ดังนี้

Tool Tip = 0

NoseZ=-NoseR*(0+IsNoseComp*sin(alpha/2+beta/2)/cos(alpha/2-beta/2));

NoseX=-NoseR*(0-IsNoseComp*cos(alpha/2+beta/2)/cos(alpha/2-beta/2));

Tool Tip = 1

NoseZ=NoseR*(1-IsNoseComp*sin(alpha/2+beta/2)/cos(alpha/2-beta/2));

NoseX=NoseR*(1+IsNoseComp*cos(alpha/2+beta/2)/cos(alpha/2-beta/2));

Tool Tip = 2

NoseZ=-NoseR*(1+IsNoseComp*sin(alpha/2+beta/2)/cos(alpha/2-beta/2));

NoseX=NoseR*(1+IsNoseComp*cos(alpha/2+beta/2)/cos(alpha/2-beta/2));

Tool Tip = 3

NoseZ=-NoseR*(1+IsNoseComp*sin(alpha/2+beta/2)/cos(alpha/2-beta/2));

NoseX=-NoseR*(1-IsNoseComp*cos(alpha/2+beta/2)/cos(alpha/2-beta/2));

Tool Tip = 4

NoseZ=NoseR*(1-IsNoseComp*sin(alpha/2+beta/2)/cos(alpha/2-beta/2));

NoseX=-NoseR*(1-IsNoseComp*cos(alpha/2+beta/2)/cos(alpha/2-beta/2));

Tool Tip = 5

NoseZ=NoseR*(1-IsNoseComp*sin(alpha/2+beta/2)/cos(alpha/2-beta/2));

NoseX=-NoseR*(0-IsNoseComp*cos(alpha/2+beta/2)/cos(alpha/2-beta/2));

Tool Tip = 6

NoseZ=-NoseR*(0+IsNoseComp*sin(alpha/2+beta/2)/cos(alpha/2-beta/2));

NoseX=NoseR*(1+IsNoseComp*cos(alpha/2+beta/2)/cos(alpha/2-beta/2));

Tool Tip = 7

NoseZ=-NoseR*(1+IsNoseComp*sin(alpha/2+beta/2)/cos(alpha/2-beta/2));

NoseX=-NoseR*(0-IsNoseComp*cos(alpha/2+beta/2)/cos(alpha/2-beta/2));

Tool Tip = 8

NoseZ=-NoseR*(0+IsNoseComp*sin(alpha/2+beta/2)/cos(alpha/2-beta/2));

NoseX=-NoseR*(1-IsNoseComp*cos(alpha/2+beta/2)/cos(alpha/2-beta/2));

เมื่อ NoseX คือ ระยะทางการชดเชยในแกน X

NoseZ คือ ระยะทางการชดเชยในแกน Z

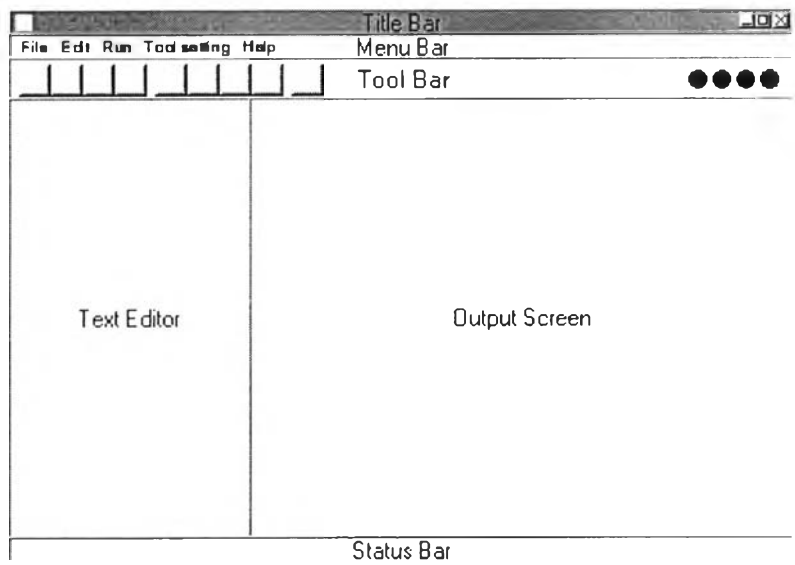
NoseR คือ รัศมีของจมูกมีด

IsNoseComp คือ ทิศทางการชดเชยจมูกมีด มีค่าเท่ากับ 1 ในโหมด G41 และ เท่ากับ –1 ในโหมด G42

3.3 การออกแบบหน้าจอโปรแกรม

หน้าจอโปรแกรมแบ่งเป็น 7 ส่วนคือ

1. ส่วนแสดงชื่อโปรแกรมและชื่อไฟล์ที่กำลังทำงาน (Title Bar)
2. แถบเมนูเพื่อควบคุมการทำงานของโปรแกรม (Menu Bar)
3. แถบปุ่มเพื่อควบคุมการทำงานของโปรแกรม (Tool Bar)
4. หน้าต่างเพื่อทำการแก้ไขข้อมูล G-Code (Text Editor)
5. หน้าต่างเพื่อแสดงผลจาก G-Code (Output Screen)
6. แถบแสดงสถานะต่างๆ (Status Bar)
7. สัญญาณไฟแสดงสถานะ (Status Lamp)

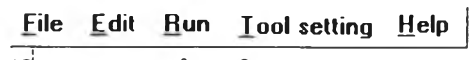


รูปที่ 3.7 แสดงส่วนประกอบของหน้าจอโปรแกรม

ส่วนที่ 1 เป็นส่วนที่แสดงชื่อโปรแกรม และชื่อไฟล์ที่กำลังทำงาน โดยชื่อไฟล์ที่กำลังทำงานจะแสดงต่อจากชื่อโปรแกรมซึ่งคั่นด้วยเครื่องหมาย " : "

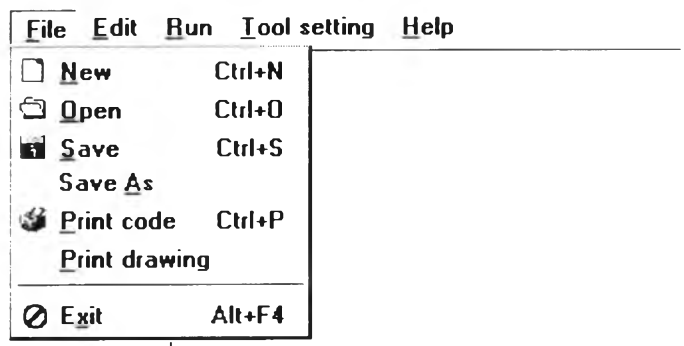
รูปที่ 3.8 แสดงหน้าจอโปรแกรมส่วนหัวโปรแกรม

ส่วนที่ 2 เป็นแถบเมนูเพื่อควบคุมการทำงานของโปรแกรม โดยประกอบด้วยเมนูหลักดังรูป



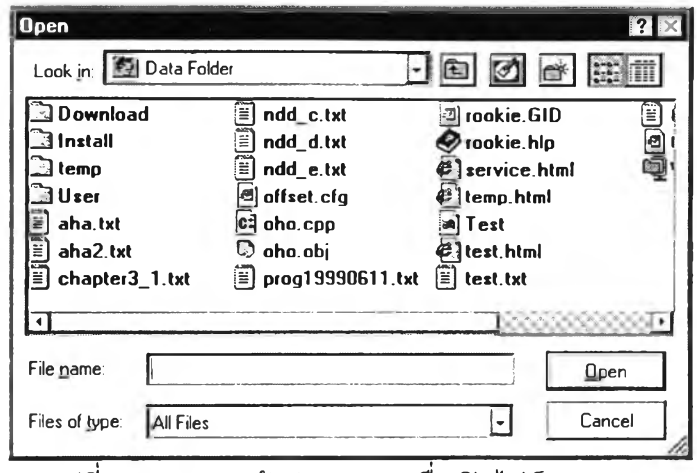
รูปที่ 3.9 แสดงหน้าจอโปรแกรมส่วนแถบเมนู

เมนู	การทำงาน
File	จัดการเกี่ยวกับไฟล์ข้อมูล
Edit	จัดการเกี่ยวกับการแก้ไขข้อความ
Run	ควบคุมการแสดงผล
Tool setting	จัดการและระบุค่าต่างๆของมีดตัด
Help	ระบบช่วยเหลือ

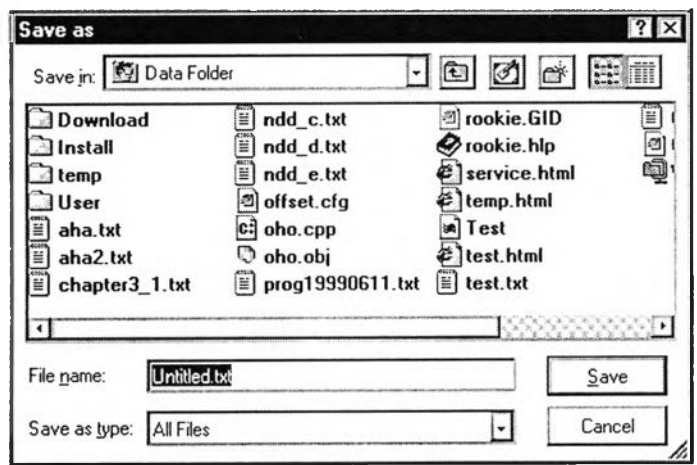


รูปที่ 3.10 แสดงรายละเอียดในเมนู File

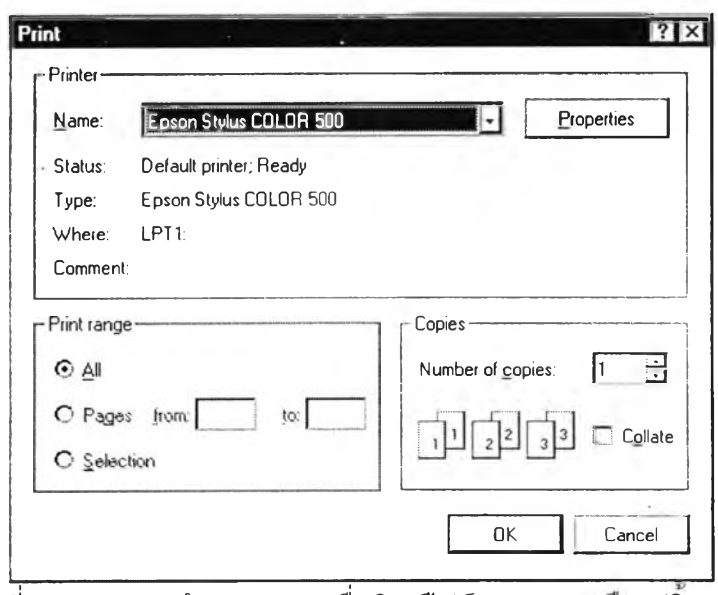
เมนู	การทำงาน
New	สร้างไฟล์ G-Code ใหม่
Open	เปิดไฟล์ G-Code เดิม
Save	บันทึกการเปลี่ยนแปลงไฟล์ G-Code
Save As	บันทึกการเปลี่ยนแปลงไฟล์ G-Code เป็นชื่อใหม่
Print Code	พิมพ์ไฟล์ G-Code ออกทางเครื่องพิมพ์
Print Drawing	พิมพ์รูปชิ้นงานออกทางเครื่องพิมพ์
Exit	ออกจากโปรแกรม



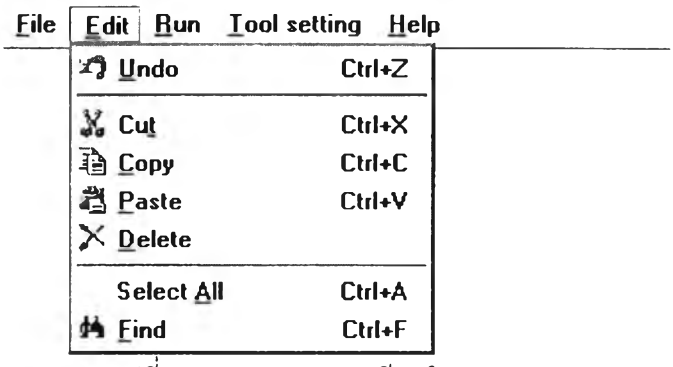
รูปที่ 3.11 แสดงหน้าต่าง open เพื่อเปิดไฟล์ G-Code



รูปที่ 3.12 แสดงหน้าต่าง saveas เพื่อบันทึกไฟล์ G-Code

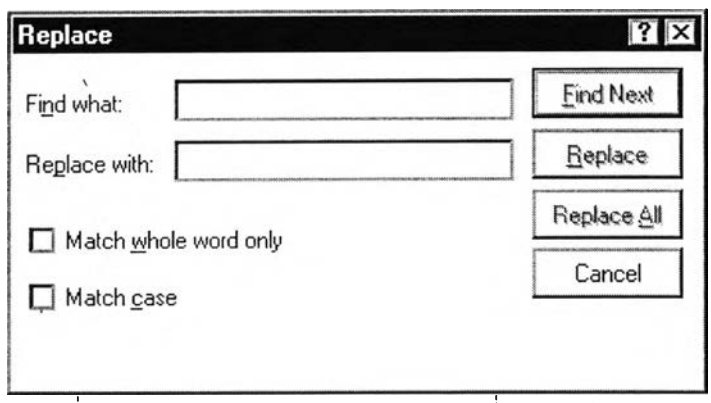


รูปที่ 3.13 แสดงหน้าต่าง print เพื่อพิมพ์ไฟล์ G-Code หรือรูปชิ้นงาน

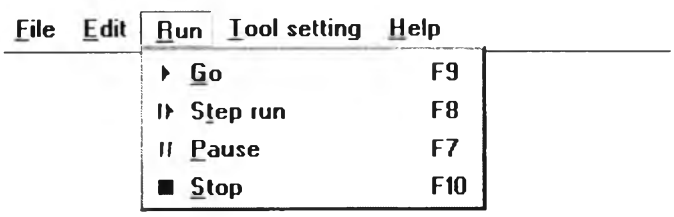


รูปที่ 3.14 แสดงรายละเอียดในเมนู Edit

เมนู	การทำงาน
Undo	ยกเลิกการแก้ไขครั้งหลังสุด
Cut	ตัดข้อความที่เลือกออก
Copy	คัดลอกข้อความที่เลือกไว้
Paste	วางข้อความที่คัดลอก
Delete	ลบข้อความที่เลือกไว้
Select All	เลือกข้อความทั้งหมด
Find	ค้นหาข้อความที่ต้องการ

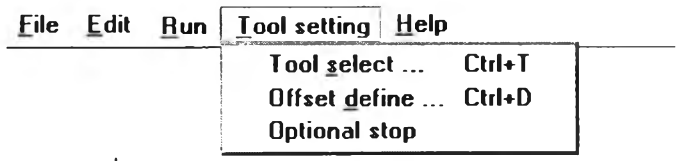


รูปที่ 3.15 แสดงหน้าต่าง find/replace เพื่อค้นหาข้อความ



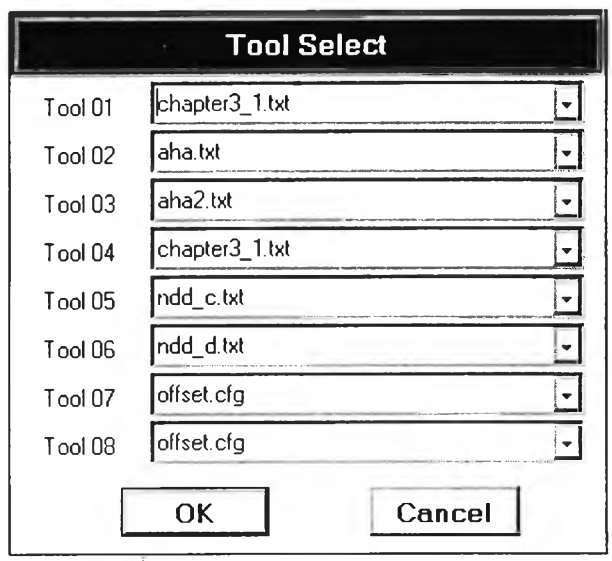
รูปที่ 3.16 แสดงรายละเอียดในเมนู Run

เมนู	การทำงาน
Go	เริ่มการจำลองการทำงานตาม G-Code ที่ระบุ
Step run	จำลองการทำงานตาม G-Code ทีละบรรทัด
Pause	หยุดการจำลองการทำงานชั่วคราว
Stop	ยกเลิกการจำลองการทำงาน

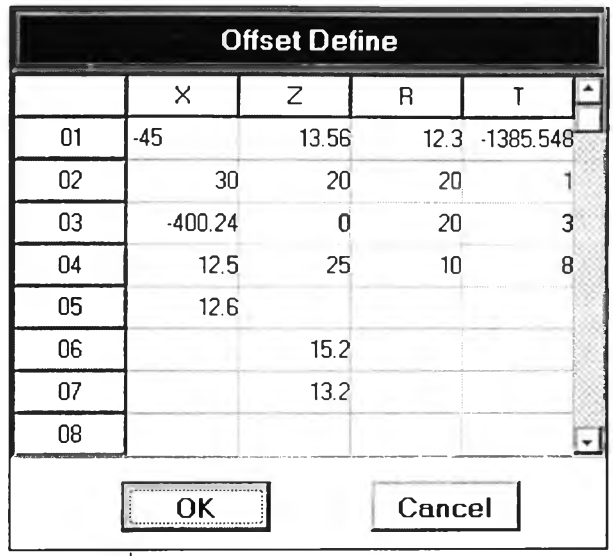


รูปที่ 3.17 แสดงรายละเอียดในเมนู Tool setting

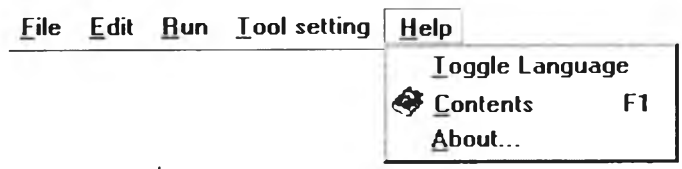
เมนู	การทำงาน
Tool select	เรียกหน้าต่างสำหรับระบุมีดตัดสำหรับหมายเลขต่างๆ
Offset define	เรียกหน้าต่างสำหรับระบุค่าออฟเซตต่างๆ
Optional stop	ทางเลือกการหยุด



รูปที่ 3.18 แสดงหน้าต่าง Tool Select



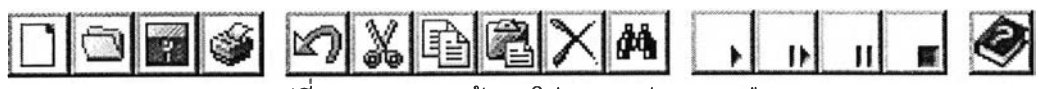
รูปที่ 3.19 แสดงหน้าต่าง Offset Define



รูปที่ 3.20 แสดงรายละเอียดในเมนู Help

เมนู	การทำงาน
Toggle Language	เปลี่ยนภาษาที่แสดงในเมนูต่างๆ
Contents	เนื้อหาของระบบช่วยเหลือ
About	เกี่ยวกับโปรแกรม

ส่วนที่ 3 เป็นแถบปุ่มสำหรับควบคุมการทำงานของโปรแกรมเช่นเดียวกับโปรแกรม โดยสร้างเป็นปุ่มสำหรับการทำงานที่เกิดขึ้นบ่อย เพื่อให้การทำงานสามารถกระทำได้อย่างสะดวกรวดเร็ว ซึ่งสามารถแสดงได้ดังรูป



รูปที่ 3.21 แสดงหน้าจอโปรแกรมส่วนแถบปุ่ม

ปุ่ม	การทำงาน
	เหมือนการทำงานของเมนู New

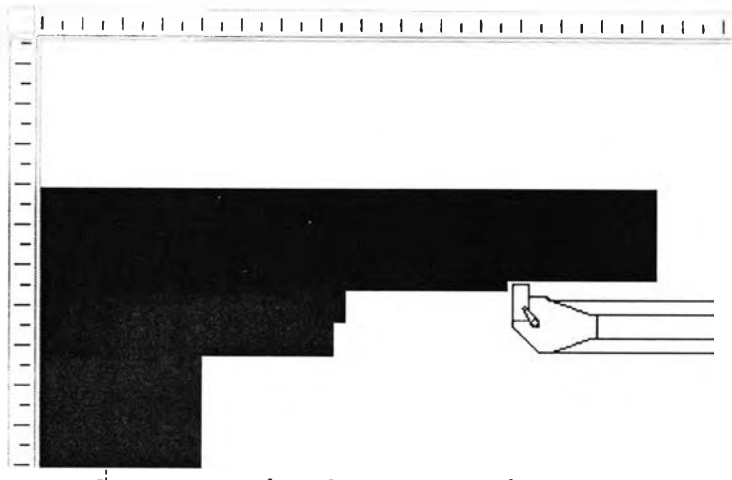
ปุ่ม	การทำงาน
	เหมือนการทำงานของเมนู Open
	เหมือนการทำงานของเมนู Save
	เหมือนการทำงานของเมนู Print แต่จะพิมพ์เลยโดยไม่แสดงหน้าต่างพิมพ์เพื่อให้กำหนดค่าต่างๆ
	เหมือนการทำงานของเมนู Undo
	เหมือนการทำงานของเมนู Cut
	เหมือนการทำงานของเมนู Copy
	เหมือนการทำงานของเมนู Paste
	เหมือนการทำงานของเมนู Delete
	เหมือนการทำงานของเมนู Find
	เหมือนการทำงานของเมนู Run
	เหมือนการทำงานของเมนู Step run
	เหมือนการทำงานของเมนู Pause
	เหมือนการทำงานของเมนู Stop
	เหมือนการทำงานของเมนู Content

ส่วนที่ 4 เป็นหน้าต่างเพื่อทำการแก้ไขข้อมูล G-Code ที่ต้องการจะจำลองการทำงาน

รูปที่ 3.22 แสดงหน้าจอโปรแกรมส่วนหน้าต่างแก้ไขข้อมูล

ส่วนที่ 5 เป็นหน้าต่างเพื่อแสดงผลการจำลองการทำงาน ประกอบด้วย

- ส่วนแสดงผล
- แถบไม้บรรทัดในแนวแกน X
- แถบไม้บรรทัดในแนวแกน Z



รูปที่ 3.23 แสดงหน้าจอโปรแกรมส่วนหน้าต่างแสดงผล

ส่วนที่ 6 เป็นแถบแสดงสถานะเพื่อบอกค่าต่างๆได้แก่

- ตำแหน่งของมีดตัดในแกน X
- ตำแหน่งของมีดตัดในแกน Z
- อัตราการป้อนตัด
- ความเร็วรอบ
- ค่าออฟเซตในแนวแกน X
- ค่าออฟเซตในแนวแกน Z
- ค่ารัศมีจมูกมีด
- ค่าระบุปลายมีดตัด
- หมายเลขของมีดตัดที่กำลังทำงาน
- วันที่ปัจจุบัน

รูปที่ 3.24 แสดงหน้าจอโปรแกรมส่วนแถบแสดงสถานะ

ส่วนที่ 7 เป็นสัญญาณไฟแสดงสถานะ ประกอบด้วย

- ไฟแสดงทิศทางการหมุน
- ไฟแสดงการกำหนดตัวเลือกการหยุด
- ไฟแสดงการใช้สารหล่อเย็น
- ไฟแสดงการตรวจสอบตำแหน่งมีดตัด
- ไฟแสดงการใช้เบรก

รูปที่ 3.25 แสดงหน้าจอโปรแกรมส่วนไฟแสดงสถานะ

3.4 การออกแบบรายงาน

เอกสารที่ได้จากโปรแกรมจะประกอบด้วยเอกสาร 2 ชนิดคือ

- เอกสารแสดงข้อมูลชุดรหัสสั่งการทำงาน โดยข้อมูลที่พิมพ์ออกมาจะได้จากส่วนหน้าต่างแก้ไขข้อมูล
- เอกสารแสดงภาพของชิ้นงาน โดยข้อมูลที่พิมพ์ออกมาจะได้จากรูปชิ้นงานที่แสดงอยู่ในส่วนหน้าต่างแสดงผล

I19267101

ชุดรหัสสั่งการทำงาน

ชื่องาน	:	วัสดุ	:	
ชิ้นงาน	:	เส้นผ่าศก.	:	มม.
ผู้เขียน	:	ความยาว	:	มม.

ตัวอย่าง เอกสารแสดงข้อมูลชุดรหัสสั่งการทำงาน (ภาษาไทย)

CNC Program Sheet

Project	:	Material	:	
Part	:	Diameter	:	mm.
Programmer	:	Length	:	mm.

ตัวอย่าง เอกสารแสดงข้อมูลชุดรหัสสั่งการทำงาน (ภาษาอังกฤษ)

ตัวอย่าง เอกสารแสดงภาพชิ้นงาน (ภาษาไทย)

ชื่องาน		
ชิ้นงาน		วัสดุ
ผู้เขียน		วันที่ 19 ตุลาคม พ.ศ. 2542

Rook.IE Lathe version 1.0

ตัวอย่าง เอกสารแสดงภาพชิ้นงาน (ภาษาอังกฤษ)

Project			
Part		Material	
Programmer		Date	October 19, 1999

Rook.IE Lathe version 1.0

3.5 โครงสร้างการเขียนโปรแกรม

ในการพัฒนาโปรแกรมประกอบด้วยไฟล์ต่างๆที่จำเป็นได้แก่

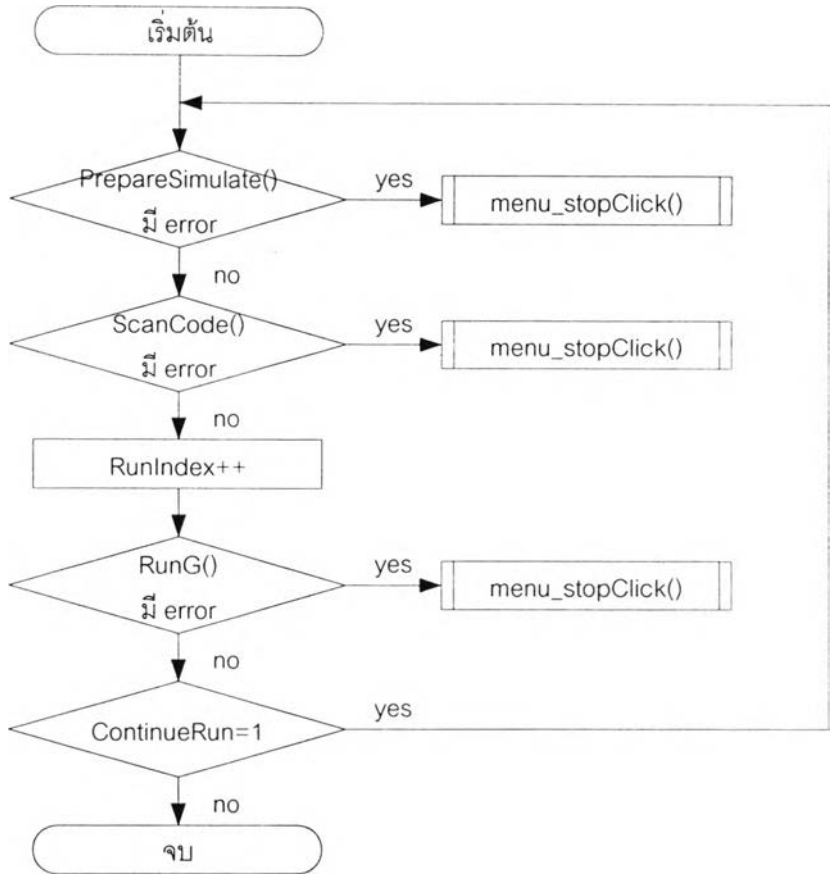
- mainmenu.bpr เก็บข้อมูลเกี่ยวกับโปรแกรมที่จะพัฒนา
- mainmenu.cpp เก็บ source code หลักของโปรแกรม
- menu.cpp เก็บ source code ของหน้าต่างหลัก
- tools.cpp เก็บ source code ของหน้าต่างระบุหมายเลขมีดตัด
- offset.cpp เก็บ source code ของหน้าต่างระบุค่าออฟเซต
- drawing.cpp เก็บ source code ของหน้าต่างพิมพ์รูปชิ้นงาน
- logo.cpp เก็บ source code ของหน้าต่างเกี่ยวกับโปรแกรม
- report.cpp เก็บ source code ของหน้าต่างพิมพ์ชุดรหัสสั่งการทำงาน
- menu.h เป็น include file ของหน้าต่างหลัก
- tools.h เป็น include file ของหน้าต่างระบุหมายเลขมีดตัด
- offset.h เป็น include file ของหน้าต่างระบุค่าออฟเซต
- drawing.h เป็น include file ของหน้าต่างพิมพ์รูปชิ้นงาน
- logo.h เป็น include file ของหน้าต่างเกี่ยวกับโปรแกรม
- report.h เป็น include file ของหน้าต่างพิมพ์ชุดรหัสสั่งการทำงาน
- menu.dfm เก็บคุณสมบัติของวัตถุ(object)ต่างๆในหน้าต่างหลัก
- tools.dfm เก็บคุณสมบัติของวัตถุ(object)ต่างๆในหน้าต่างระบุหมายเลขมีดตัด
- offset.dfm เก็บคุณสมบัติของวัตถุ(object)ต่างๆในหน้าต่างระบุค่าออฟเซต
- drawing.dfm เก็บคุณสมบัติของวัตถุ(object)ต่างๆในหน้าต่างพิมพ์รูปชิ้นงาน
- logo.dfm เก็บคุณสมบัติของวัตถุ(object)ต่างๆในหน้าต่างเกี่ยวกับโปรแกรม
- report.dfm เก็บคุณสมบัติของวัตถุ(object)ต่างๆในหน้าต่างพิมพ์ชุดรหัสสั่งการทำงาน

ในส่วนของการควบคุมการจำลองการทำงานประกอบด้วย ฟังก์ชัน 4 ฟังก์ชันคือ

- menu_goClick
- menu_stepClick
- menu_pauseClick
- menu_stopClick

โดยที่ฟังก์ชัน menu_stepClick จะเป็นฟังก์ชันหลักในการจำลองการทำงาน และมีตัวแปรชื่อ ContinueRun เพื่อควบคุมการจำลองการทำงานซึ่งจะมีค่าเป็น 0 เมื่อเป็นการจำลองทำงานแบบทีละ

บรรทัด และเป็น 1 เมื่อเป็นการจำลองการทำงานแบบต่อเนื่อง และมีตัวแปรชื่อ RunIndex เพื่อระบุบรรทัดที่จะทำการจำลองการทำงาน โดยการทำงานของฟังก์ชัน menu_stepClick สามารถแสดงได้ดังรูปที่ 3.7



รูปที่ 3.7 แสดงการทำงานของฟังก์ชัน menu_stepClick

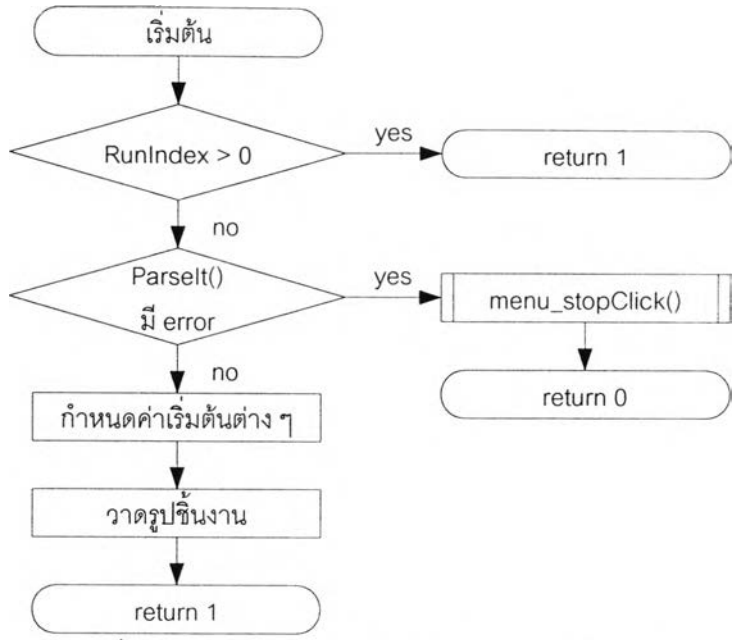
สำหรับฟังก์ชัน menu_goClick จะทำการกำหนดให้ตัวแปร ContinueRun มีค่าเป็น 1 แล้วจึงทำการเรียกฟังก์ชัน menu_stepClick ให้ทำงาน

ฟังก์ชัน menu_pauseClick จะทำการกำหนดให้ตัวแปร ContinueRun มีค่าเป็น 0

และ ฟังก์ชัน menu_stopClick จะทำการหยุดการจำลองการทำงานทั้งหมด

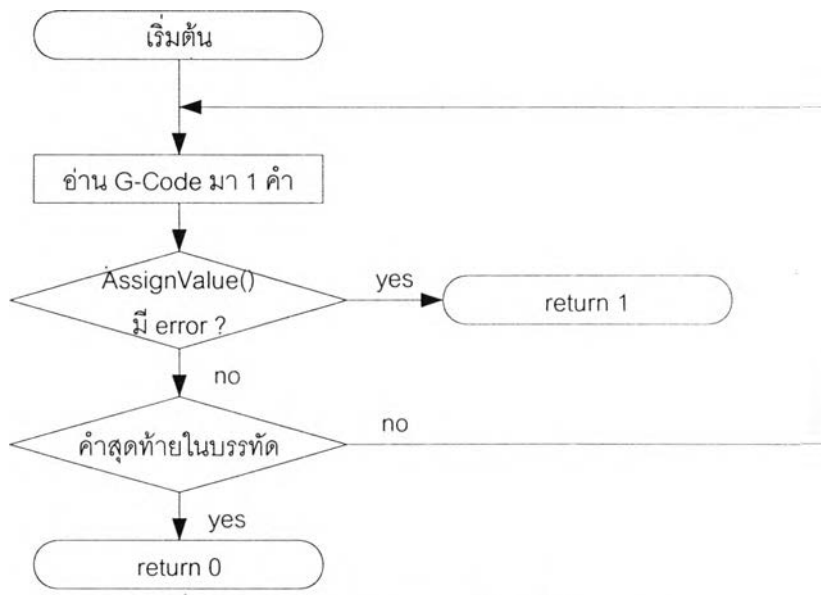
นอกจากฟังก์ชันทั้ง 4 ที่ใช้ในการควบคุมการจำลองการทำงานแล้ว ยังมีฟังก์ชันที่สำคัญสำหรับการจำลองการทำงานได้แก่

- ฟังก์ชัน PrepareSimulation ทำการเตรียมพร้อมสำหรับการจำลองการทำงาน โดยจะส่งค่า 0 ออกมาถ้ามีข้อผิดพลาดเกิดขึ้น เช่น มีรหัสสั่งการทำงานที่ผิดไวยากรณ์ และส่งค่า 1 เมื่อไม่มีข้อผิดพลาดเกิดขึ้น โดยจะมีการทำงานดังรูปที่ 3.8



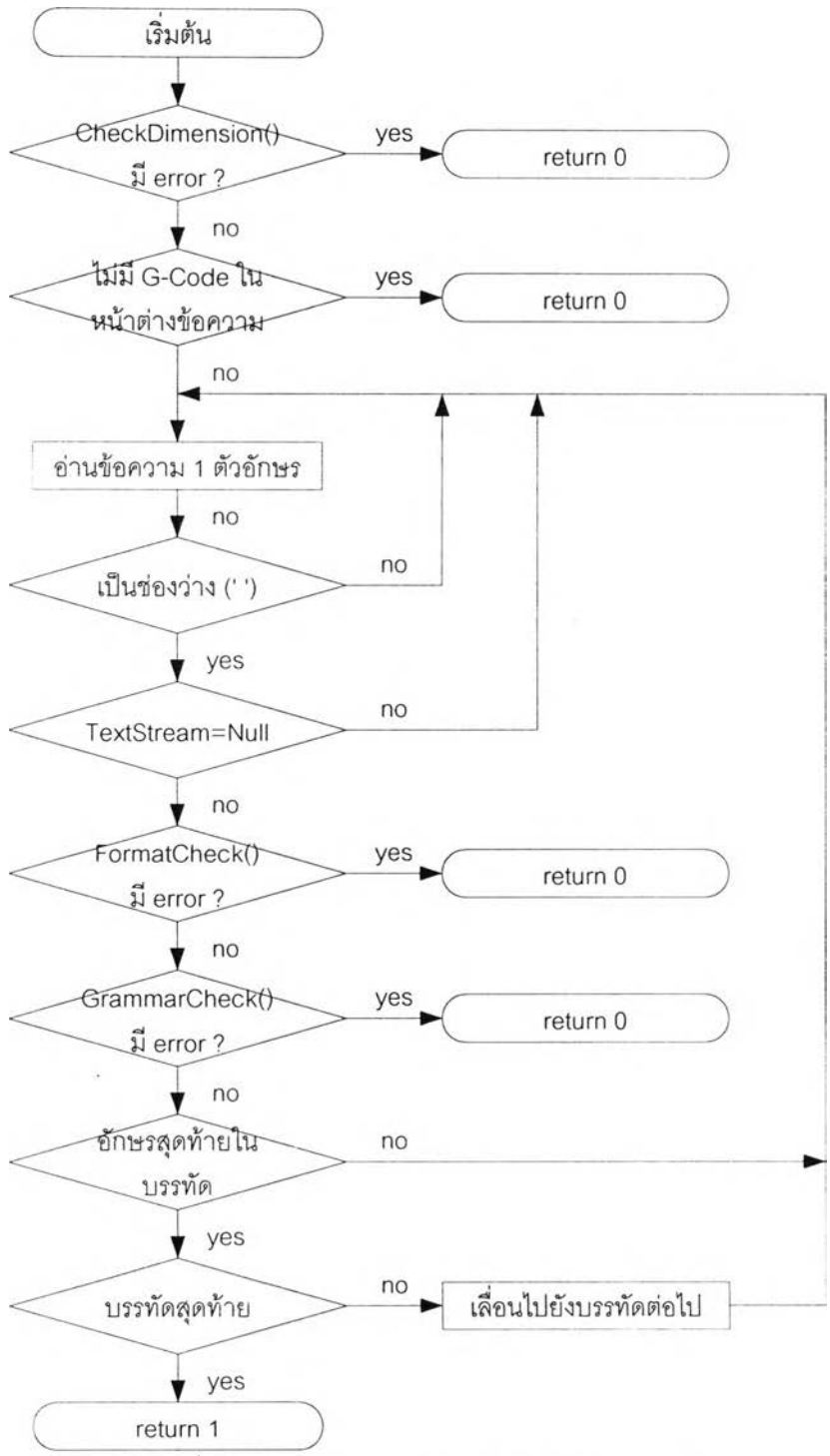
รูปที่ 3.8 แสดงการทำงานของฟังก์ชัน PrepareSimulate

- ฟังก์ชัน ScanCode เป็นฟังก์ชันที่ทำการอ่านรหัสสั่งการทำงานในบรรทัดที่ระบุด้วยตัวแปร RunIndex เข้ามาทีละ 1 คำ แล้วทำการระบุค่าให้แก่ตัวแปรต่างๆเพื่อทำการจำลองการทำงานต่อไป จะให้ค่า 0 เมื่อไม่มีข้อผิดพลาด และให้ค่า 1 เมื่อเกิดข้อผิดพลาด ดังแสดงในรูปที่ 3.9



รูปที่ 3.9 แสดงการทำงานของฟังก์ชัน ScanCode

- ฟังก์ชัน ParseIt ทำการอ่านรหัสสั่งการทำงานเข้ามาทีละคำจนจบชุดรหัสสั่งการทำงาน เพื่อทำการตรวจสอบไวยากรณ์ คืนค่า 0 เมื่อมีข้อผิดพลาด และคืนค่า 1เมื่อไม่มีข้อผิดพลาด ดังรูปที่ 3.10



รูปที่ 3.10 แสดงการทำงานของฟังก์ชัน ParseIt

- ฟังก์ชัน FormatCheck จะรับรหัสสั่งการทำงานเข้ามา 1 คำและทำการตรวจสอบรูปแบบของอักขระและค่าตัวเลขที่ตามมา ถ้ามีข้อผิดพลาดจะคืนค่า 1 และ คืนค่า 0 เมื่อไม่มีข้อผิดพลาด
- ฟังก์ชัน GrammarCheck จะรับรหัสสั่งการทำงานเข้ามา 1 คำและทำการตรวจสอบไวยากรณ์ของรหัสสั่งการทำงานในบรรทัดนั้นเช่น มีอักระ X มากกว่า 1 ตัวในบรรทัดเดียวกันเป็นต้น ถ้ามีข้อผิดพลาดจะคืนค่า 1 และ คืนค่า 0 เมื่อไม่มีข้อผิดพลาด
- ฟังก์ชัน AssignValue เป็นฟังก์ชันที่ทำการระบุค่าให้แก่ตัวแปรต่างๆเพื่อทำการจำลองการทำงานต่อไป
- ฟังก์ชัน RunG เป็นฟังก์ชันที่ทำการควบคุมการเรียกฟังก์ชันที่แสดงการเคลื่อนไหวให้ตรงกับรหัสสั่งการทำงานในแต่ละบรรทัด