บทที่ 7

การวิเคราะห์การป้องกันแบบประสานการทำงานด้วยโปรแกรมคอมพิวเตอร์

7.1 คำนำ

การประยุกต์ใช้งานแอพพลิเคชันบน Microsoft windows นั้นให้ทั้งความตื่นตาตื่นใจและความง่ายในการใช้งานต่อผู้ใช้ ด้วยการติดต่อกับผู้ใช้แบบกราฟฟิก ซึ่งมีทั้งภาพ แสง และ สี ที่ชวนให้สนใจมากกว่าการใช้งานในรูปแบบของเท็กซ์โหมดบนระบบปฏิบัติการ DOS แต่เบื้องหลังในการพัฒนาแอพพลิเคชันขึ้นมานั้น มีเพียงผู้เขียนโปรแกรมเท่านั้นที่รู้ว่ายุ่งยากและลำบากเพียงใด โดยเฉพาะอย่างยิ่งในสมัยเริ่มแรกที่มีเพียงภาษา C และชุด Software development kit (SDK) ของ Microsoft เท่านั้น ผู้พัฒนาจะต้องมีความรู้ความเข้าใจในภาษาและการทำงานของ Windows เป็นอย่างดี ที่สำคัญคือต้องรู้และเขียนโปรแกรมด้วยภาษา C ได้อย่างคล่องแคล่ว รวมทั้งต้องเข้าใจการใช้พอยเตอร์และหลักการของภาษาอย่างถ่องแท้ ซึ่งสิ่งเหล่านี้นับว่าเป็นเรื่องที่ลำบากมากสำหรับโปรแกรมเมอร์สมัครเล่น ในการที่จะสร้างแอพพลิเคชันสำหรับ Windows ขึ้นมา

ที่กล่าวมานั้น เป็นยุคแรกของการพัฒนาโปรแกรมบน Windows ต่อมารูปการณ์เริ่มเปลี่ยนไป เมื่อบริษัท Borland International ได้นำผลิตภัณฑ์ที่ชื่อ Turbo pascal for windows ซึ่งให้ความง่ายในการเขียนโปรแกรมบน Windows ด้วยลักษณะคำสั่งและการใช้งานเหมือนกับ Turbo pascal บน DOS ที่มีผู้นิยมใช้กันมากและมีความคุ้นเคยกันเป็นอย่างดี ซึ่งถือได้ว่าเป็นการลดความซับซ้อนในการพัฒนาแอพพลิเคชันบน Windows ลงมาได้ระดับหนึ่ง นอกจากนี้ยังมีซอฟต์แวร์อีกหลายตัวด้วยกัน ที่มาช่วยในการสร้างแอพพลิเคชันบน Windows เช่น โปรแกรม Toolbooks ที่ทำงานในลักษณะของ Program generator คือผู้ใช้จะเป็นผู้ออกแบบเมนู, องค์ประกอบต่างๆในวินโดว์ และกำหนดเงื่อนไขในการทำงาน หลังจากนั้นซอฟต์แวร์พวกนี้จะสร้างโปรแกรมในภาษา C มาให้เพื่อให้ผู้ใช้แก้ไขและคอมไพล์เป็นไฟล์ที่ทำงานได้ต่อไป

ทั้ง Turbo pascal for windows, Toolbooks, และกลุ่ม Program generator ต่างก็เป็นเครื่องมือที่เป็นทางเลือกอื่นนอกเหนือจากภาษา C และชุด SDK ไม่ว่าจะเป็น Turbo pascal for Windows หรือ Toolbooks ซึ่งต่างก็ปลดตัวเองจากพันธนาการของ SDK นอกจากนั้นทุกคนคงจะไม่ลืมว่าหนึ่งในภาษาแรกๆ ที่ผู้สนใจการเขียนโปรแกรมมักจะเรียนกัน ก็คือภาษา Basic ซึ่งมีจุดเด่นที่ความง่าย สะดวกต่อการเขียน และการเรียนรู้ โดยทาง Microsoft เองก็รู้ถึงจุดนี้เป็นอย่างดี

อีกทั้งยังรู้ว่าแนวโน้มของผู้ใช้จะนิยมพัฒนาโปรแกรมขึ้นมาใช้เองมีมากขึ้น หรือแม้แต่นักพัฒนามืออาชีพก็คงต้องการความสะดวกและง่ายในการพัฒนาแอพพลิเคชันบน Windows มากขึ้น ดังนั้นจึงไม่แปลกที่ Microsoft จะมีตัวแปลภาษา BASIC สำหรับการพัฒนาแอพพลิเคชันบน Windows โดยให้ชื่อว่า Visual Basic ที่ง่ายต่อการใช้งาน ขณะเดียวกันก็ง่ายต่อการเขียนโปรแกรมตามแบบฉบับของภาษา BASIC ด้วย

Visual Basic เวอร์ชันแรกที่ออกมานั้นเป็นเวอร์ชันบน Windows ซึ่งออกสู่สายตาผู้ใช้เมื่อปี 1991 ในเวอร์ชัน 1.0 นั้นยังไม่มีความสามารถอะไรมากนัก เป็นเพียงเครื่องมืออย่างง่ายสำหรับการสร้างแอพพลิเคชันบน Windows มากกว่า องค์ประกอบหรือ Object ที่ใช้ได้ก็มีแค่องค์ประกอบพื้นฐานของ Windows เช่น Text box, List box เท่านั้น ซึ่งแม้แต่การใช้งานแบบ Multiple document interface (MDI) ซึ่งหมายถึงการมีวินโดว์ย่อยๆในวินโดว์หลัก ก็ไม่สามารถทำได้ แต่อย่างไรก็ตาม นับว่า Visual Basic 1.0 ประสบความสำเร็จและได้รับการต้อนรับจากผู้ใช้เป็นอย่างดี ในแง่ของการใช้และหลักการโปรแกรมแบบ Visual จนทาง Microsoft ต้องทำ Visual Basic for DOS ออกมาด้วย ในแต่ละเวอร์ชันก็ได้มีการพัฒนาปรับปรุงความสามารถขึ้นเรื่อยๆ เช่น สนับสนุน MDI, OLE และมี Object มากขึ้น

ในอนาคตคาดว่า Visual Basic จะมีบทบาทต่อผู้ใช้ Windows มากขึ้น [3] เพราะต่อไปแอพพลิเคชันของ Microsoft ทุกตัวบน Windows จะมีภาษามาโครเดียวกันหมด คือ Visual Basic for application (VBA) นั่นคือ แทนที่ผู้ใช้จะต้องเรียนรู้ภาษามาโครของ Word for Windows, EXCEL หรือโปรแกรมอื่นของ Microsoft แต่ละตัว ก็จะเป็นการเรียนรู้ VBA เพียงอย่างเดียว ทั้งนี้เนื่องจากความง่ายของภาษาสำหรับแอพพลิเคชันบน Windows อีกด้วย ในลักษณะเดียวกันกับภาษา REXX ของ OS/2 นอกจากนี้ก็ยังมีข่าวการพัฒนา Visual Basic บนรูปแบบอื่นออกมาเป็นระยะๆ ด้วย ฉะนั้นจึงถือได้ว่าการเรียนรู้ Visual Basic นั้นจะทำให้ผู้ใช้ก้าวไปสู่การใช้แอพพลิเคชันสำเร็จรูป อย่างมีประสิทธิภาพมากขึ้น นอกเหนือจากการพัฒนาแอพพลิเคชันขึ้นมาใช้เอง อย่างไรก็ตาม VBA จะเป็นเพียงการดึงบางคำสั่งและฟังก์ชันของ Visual Basic มาใช้เท่านั้น

7.2 ภาพรวมของ Visual Basic

Visual Basic มีสภาพแวดล้อมสำหรับการพัฒนาบนโปรแกรมบน Windows ประกอบด้วยเครื่องมือต่างๆ อย่างครบถ้วน ไม่ว่าจะเป็นส่วนของการออกแบบ, ส่วนแสดงผลของผู้ใช้ (User interface), ส่วนออกแบบเมนู (Menu design), การสร้างรายงาน (Report writer), อิดิเตอร์สำหรับ

ป้อนโปรแกรม และส่วนที่ใช้ในการตรวจหาข้อผิดพลาดในโปรแกรม (Debugger) องค์ประกอบเหล่านี้นับว่าเอื้ออำนวยต่อการทำงานของโปรแกรมเมอร์เป็นอย่างมาก

ในด้านของตัวภาษา Visual Basic ได้มีการนำหลักการของ BASICA และ GW-BASIC [2-3, 24] มาใช้ โดยสนับสนุนความสามารถเดิมเกือบทั้งหมด นอกจากนี้ยังได้เพิ่มการโปรแกรมแบบมีโครงสร้างของ QuickBasic ซึ่งคล้ายกับในภาษาที่มีโครงสร้าง เช่น Pascal เข้าไปด้วย

นอกจากนี้ยังมีการเพิ่มคำสั่งและฟังก์ชันเกี่ยวกับ Object และการเรียกฟังก์ชันของระบบปฏิบัติการ (API) เพื่อให้การทำงานกว้างขวางขึ้น รวมทั้งสนับสนุนความสามารถของระบบ เช่น OLE, DDE และการใช้คลิปบอร์ด เป็นต้น

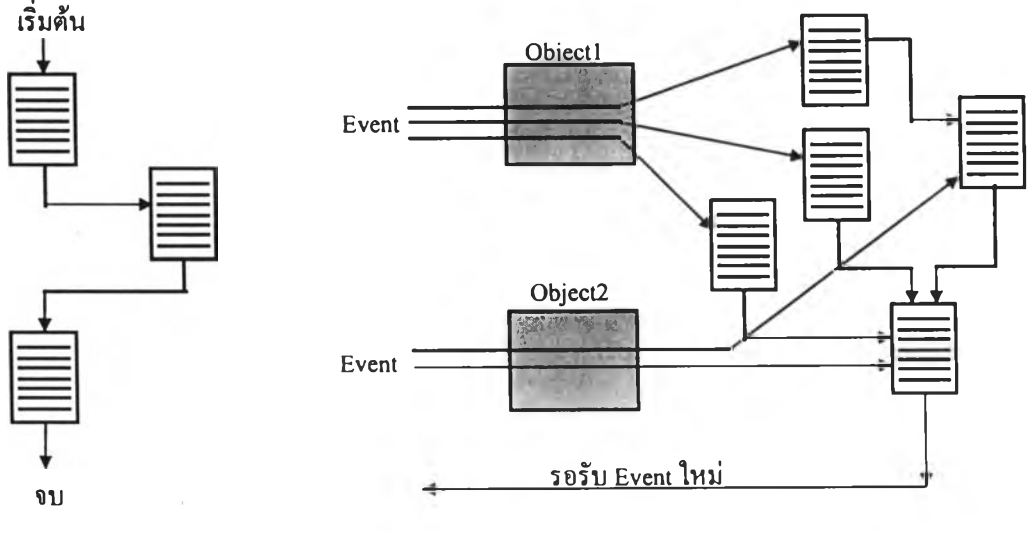
Visual Basic แต่ละเวอร์ชันจะมี 2 Edition คือ Standard และ Professional edition ซึ่งมีข้อแตกต่างกัน คือ ในชุด Professional นั้นจะมี Custom control (Object ที่สามารถนำมาใช้ในฟอร์ม) มากกว่า และจะมีเครื่องมืออื่นๆ เช่น Help compiler สำหรับการสร้างข้อความอธิบายการใช้โปรแกรม แต่ในด้านความสามารถของภาษาแล้ว จะเหมือนกันทั้ง 2 Edition ซึ่งจากความสามารถของ Visual Basic ผู้ใช้โปรแกรมสามารถนำไปสร้างแอพพลิเคชันได้มากมายหลากหลายประเภท ไม่ว่าจะเป็นโปรแกรมกราฟฟิก, โปรแกรมแสดงข้อมูล หรือแม้แต่โปรแกรมการใช้งานทางวิศวกรรมศาสตร์

7.3 หลักการโปรแกรมเชิงภาพของ Visual Basic [3]

ใน Visual Basic นั้นการพัฒนาและเขียนโปรแกรมจะเป็นไปในอีกรูปแบบหนึ่ง กล่าวคือ ในการเขียนโปรแกรมแบบเดิมนั้นเราจะต้องมานั่งออกแบบหน้าจอ ระบุตำแหน่งการแสดงผล คิดหาขั้นตอนการทำงานและอื่นๆ จากนั้นจึงทำการเขียนโปรแกรม โปรแกรมที่ได้จะอธิบายและสั่งงานคอมพิวเตอร์เป็นลำดับไป แต่ใน Visual Basic จะใช้หลักของภาพและการมองเห็น โดยเริ่มจากออกแบบวินโดว์ย่อยหรือที่ใน Visual Basic เรียกว่า ฟอร์ม (Form) โดยในแต่ละฟอร์มจะประกอบด้วยสิ่งต่างๆ ที่เราจะทำงานด้วยหรือเรียกว่าเป็น Object เช่น ข้อความ, ช่องรับข้อความ, แถบเลื่อน (Scroll bar) หรือปุ่ม (Button) ซึ่งเมื่อกำหนดสิ่งเหล่านี้ครบตามความต้องการแล้วจึงระบุว่าองค์ประกอบแต่ละอย่างจะทำงานอย่างไร โดยเขียนโปรแกรมย่อยๆ ปะเข้าไปกับ Object เหล่านั้น ที่ต้องทำแบบนี้ก็เพราะว่าการทำงานใน Windows เป็นแบบที่เรียกว่า Event-driven ซึ่งหมายถึง การทำงานของโปรแกรมที่ขึ้นกับเหตุการณ์ (Event) การเขียนโปรแกรมแบบเดิมคือสั่งงานตามลำดับ จะยุ่งยากมากซึ่งบางกรณีอาจทำไม่ได้เลย เพราะในขณะใดขณะหนึ่งนั้น ในระบบ

ไม่ใช่จะมีเพียงแอพพลิเคชันเดียวเท่านั้นที่ทำงานอยู่ Windows จะต้องจัดการกับทุกแอพพลิเคชันที่ทำงานในขณะนั้นทั้ง หมดไปพร้อมๆกัน (เนื่องจากเป็นระบบแบบ Multitasking หรือทำงานได้หลายงานในเวลาเดียวกัน) ในขณะที่โปรแกรมแสดงหน้าจอสำหรับรับอินพุต ผู้ใช้อาจพิมพ์ข้อมูลเข้าไป หรืออาจใช้เมาส์เลื่อนไปคลิ๊กตรงนั้นตรงนี้ได้ตามอิสระ ทำให้ยากที่จะเขียนโปรแกรมธรรมดาให้คอยดักเส้นทางการทำงานในการรับอินพุตว่าจะเกิดอะไรขึ้นตรงไหนได้ จึงต้องใช้รูปแบบการโปรแกรมในลักษณะ Event-driven ดังกล่าว ซึ่ง Object แต่ละตัว ก็จะมีเหตุการณ์เกิดขึ้นกับ Object นั้นๆได้หลายอย่าง ซึ่งถ้าผู้ใช้สนใจเฉพาะเหตุการณ์ใดก็เขียนโปรแกรมสั่งงานให้คอยดักหรือทำงานตามเฉพาะเหตุการณ์นั้นๆได้ เช่น ถ้าสิ่งที่ผู้ใช้สนใจเป็นปุ่มควบคุมและต้องการให้ทำงานเมื่อคลิ๊กหรือดับเบิลคลิ๊ก ก็เพียงระบุว่า หากมีการคลิ๊กที่ปุ่มควบคุมนี้โปรแกรมจะต้องทำอย่างไร หรือถ้ามีการดับเบิลคลิ๊กจะต้องทำอย่างไร ส่วนเหตุการณ์อื่นๆ ที่ไม่ได้ระบุไว้ก็จะไม่มีผลต่อ Object นั้น

จากแนวทางนี้จะเห็นว่ามีลักษณะเป็นธรรมชาติมาก คือกำหนดหน้าตาของจอภาพหรือส่วนติดต่อกับผู้ใช้ และระบุว่าถ้าเกิดเหตุการณ์อย่างนี้กับสิ่งนี้จะต้องทำอย่างไร ซึ่งโปรแกรมที่ผู้ใช้จะเขียนกันนี้ก็คือส่วนที่จะบอกว่า "จะต้องทำอย่างไร" นั่นเอง



รูปที่ 7.1 การเขียนโปรแกรมแบบธรรมดากับแบบ Event-driven

7.4 ลักษณะของโปรแกรม

จากที่ได้กล่าวไว้แล้วในส่วนของขั้นตอนและวิธีดำเนินการในบทที่ 1 ถึงโครงสร้างของโปรแกรมซึ่งได้ถูกแบ่งเป็น 3 ส่วน คือ

1) ส่วนที่ใช้สร้างไดอะแกรมเส้นเดี่ยว (Single-line diagram)

2) ส่วนคำนวณหาค่ากระแสลัดวงจรของระบบจ่ายพลังงานไฟฟ้าแบบเรเดียลภายในอาคาร (Fault-current calculation)

3) ส่วนวิเคราะห์และออกแบบระบบไฟฟ้าโดยใช้ เส้นโค้งลักษณะการทำงานของอุปกรณ์ป้องกัน (Coordinative analysis)

โดยในแต่ละส่วนนี้จะมีลักษณะบางอย่างที่เหมือนกัน คือ

- ยึดรูปแบบ รวมทั้งคำสั่งมาตรฐานของโปรแกรมบนระบบ Windows

- ส่วนอธิบายการใช้งานของโปรแกรม (Help) จะแยกอธิบายในโปรแกรมแต่ละส่วน โดยข้อความใน Help จะมีทั้งภาษาไทยและภาษาอังกฤษ

- มีเสียง "Beep" เมื่อผู้ใช้โปรแกรมกดปุ่มบางคำสั่ง

- มีการแสดงเวลา และวันที่บนหน้าจอเสมอ ขณะที่ทำการออกแบบระบบป้องกัน

- มีการเปลี่ยนรูปของเมาส์ จากแบบลูกศรไปเป็นแบบต่างๆ แล้วแต่ตำแหน่งและหน้าที่

- มีการเปลี่ยนสีของฟอร์ม (Form) ที่ใช้จากสีขาวให้เป็นสีในเชิงสีเทาทั้งหมด เพื่อความสวยงาม และความสบายตาแก่ผู้ใช้โปรแกรม

ในการเขียนโปรแกรมด้วย Visual Basic # 3 ผู้จัดทำได้ใช้ตัวเลือก (Option) และคำสั่งต่างๆ ที่เหมาะสมกับหน้าที่และจุดประสงค์ของโปรแกรม ซึ่งนอกจาก Object พื้นฐานที่ต้องใช้ เช่น Text box, Panel, Label, Picture box, Image box หรือ Command button ผู้จัดทำก็ยังเพิ่ม Object และ Option อื่นๆอีก คือ

- ใช้ 3D command button และ 3D panel ซึ่งเป็นลักษณะของ Object แบบ 3 มิติ เพื่อเพิ่มความสวยงาม และเน้นรายละเอียดให้กับรูปแบบของโปรแกรม

- ใช้ Menu design window ร่วมกับ Common dialog box object ในการออกแบบคำสั่งบน Menu bar ให้มีลักษณะแบบ Pull-down และสามารถเลือกคำสั่งโดยกดปุ่ม ALT แล้วตามด้วยตัวอักษรที่ขีดเส้นใต้ในคำสั่งได้ เช่น กด ALT และปุ่ม S ในคำสั่ง Save

- มีการใช้ Object พิเศษ ในการเชื่อมโยงข้อมูลกับโปรแกรมอื่นภายใน Windows ซึ่งในส่วนนี้ได้ใช้เชื่อมกับโปรแกรมจัดทำเอกสาร Microsoft Word ในการถ่ายโอนข้อมูลที่ใช้ในส่วน Help ของโปรแกรม

ในหัวข้อต่อไป จะกล่าวถึงลักษณะของโปรแกรมที่พัฒนาขึ้นมาอย่างละเอียด พร้อมทั้งการใช้งานโปรแกรมในลำดับถัดไป

7.4.1 **ส่วนที่ใช้ในการสร้างไดอะแกรมเส้นเดียว** (Single-line diagram)

วัตถุประสงค์ของโปรแกรมในส่วนนี้ มุ่งหวังให้ผู้ใช้สามารถออกแบบ และสร้างไดอะแกรมเส้นเดี่ยวได้อย่างสะดวกรวดเร็ว พร้อมทั้งสามารถจัดเก็บงานเก่า และพิมพ์ข้อมูลที่จำเป็นเพื่อใช้เป็นเอกสารอ้างอิง (Reference) กับโปรแกรมในส่วนต่อไป โดยจะมีรายละเอียดของโปรแกรมดังต่อไปนี้

7.4.1.1 **ลักษณะของโปรแกรมส่วนที่** 1

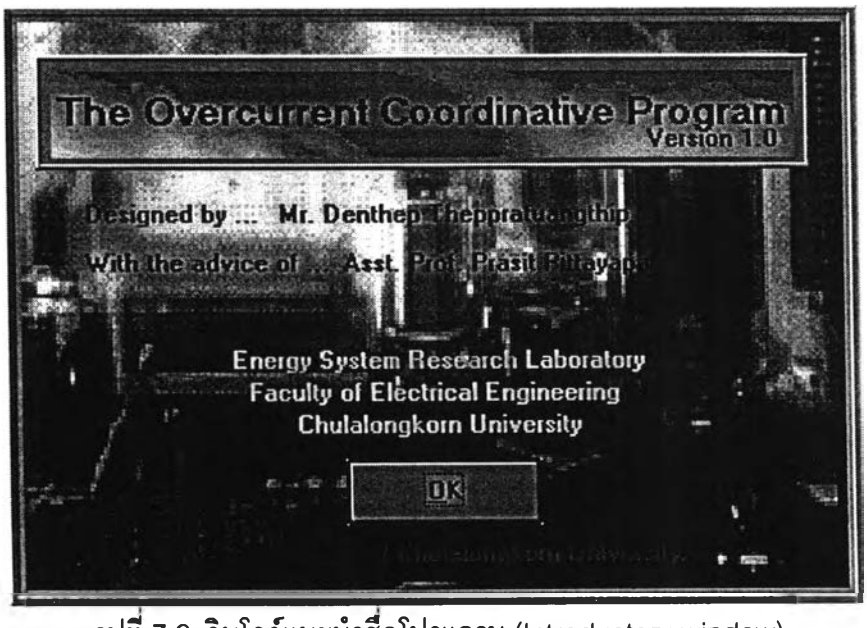
- มีวินโดว์แนะนำชื่อโปรแกรม (Introductory window) ก่อนการใช้งาน
- สามารถเปิดแฟ้มงานเก่า (Open) ที่มีข้อมูลเดิมอยู่แล้วเพื่อแก้ไข
- สามารถเริ่มงานใหม่ได้ โดยใช้คำสั่ง New ในแถบเมนู (Menu bar)
- สามารถจัดเก็บงานเก่าได้ โดยมีให้เลือกทั้งการจัดเก็บงานแบบชื่อเดิม (Save) และจัดเก็บงานเก่าแบบเปลี่ยนชื่อใหม่ (Save As)
- ภายในคำสั่งเปิดแฟ้มงานเก่า (Open), จัดเก็บงาน (Save, Save As) และ พิมพ์งาน (Print) ได้จัดให้มีรูปแบบมาตรฐาน
- สามารถออกจากโปรแกรมเพื่อไปใช้โปรแกรมอื่น โดยใช้คำสั่ง Exit บนแถบเมนู (Menu bar)
- มีส่วนช่วยอธิบายการใช้งานของโปรแกรม (Help) เป็นภาษาอังกฤษ
- มีการจัดเก็บข้อมูลที่จำเป็นบางอย่างให้ผู้ใช้ เช่น ตารางแสดงค่าความต้านทาน และค่ารีแอกแตนซ์ต่อหน่วยความยาวของสายไฟ เป็นต้น
- ผู้ใช้โปรแกรมสามารถสร้างรูปอุปกรณ์, ลากเส้นบัส (Bus) และกำหนดจุดผิดพร่อง (Fault) บนไดอะแกรมได้โดยสะดวก
- สามารถลบรูปภาพ และเส้นบัสที่สร้างขึ้นมาได้ ซึ่งมีตัวเลือกให้ลบเฉพาะรูป (Delete) หรือลบรูปที่สร้างขึ้นมาทั้งหมด (Clear)
- มีการเชื่อมต่อกับโปรแกรมในส่วนอื่น ซึ่งผู้ใช้สามารถเข้าสู่โปรแกรมในส่วนที่ 2 โดยกดปุ่ม Fault Calculation และเข้าสู่โปรแกรมในส่วนที่ 3 โดยกดปุ่ม Next
- หลังจากสร้างไดอะแกรมเสร็จแล้ว ผู้ใช้โปรแกรมสามารถลงรายละเอียดของอุปกรณ์ผ่านทางวินโดว์ข้อมูล พร้อมทั้งกำหนดจุดผิดพร่องบนไดอะแกรมได้ ซึ่งรายละเอียดทั้งสองนี้จะถูกพิมพ์ออกมาพร้อมกับไดอะแกรมที่สร้างขึ้น

- มีการแสดงข้อความใน Message box เพื่อถามย้ำผู้ใช้โปรแกรมว่าต้องการใช้ปุ่มคำสั่งนั้นหรือไม่ เพราะจะทำให้โปรแกรมทั้งหมดถูกลบไปโดยไม่มีการจัดเก็บ ซึ่งได้แก่ปุ่มลบรูปภาพทั้งหมด (Clear) และปุ่มออกจากโปรแกรม (Exit)

- มีการสร้างรูปภาพจากโปรแกรมกราฟฟิก รวมทั้งมีการใช้ (Load) ภาพที่สร้างขึ้นจากโปรแกรมอื่น หรือที่มีอยู่ในโปรแกรม Windows เพื่อนำเอาลักษณะพิเศษของรูปในโปรแกรมนั้นๆ มาใช้

7.4.1.2 รูปแบบของโปรแกรมส่วนที่ 1

โปรแกรมจะเริ่มการทำงานด้วยการแสดง Introductory window ที่ใช้ในการแนะนำชื่อของโปรแกรม คือ "Overcurrent Coordinative Program" ซึ่งจะมีปุ่มกด OK เมื่อพร้อมที่จะเข้าไปในส่วนการสร้างไดอะแกรม โดย Introductory window นี้จะมีลักษณะดังรูปที่ 7.2

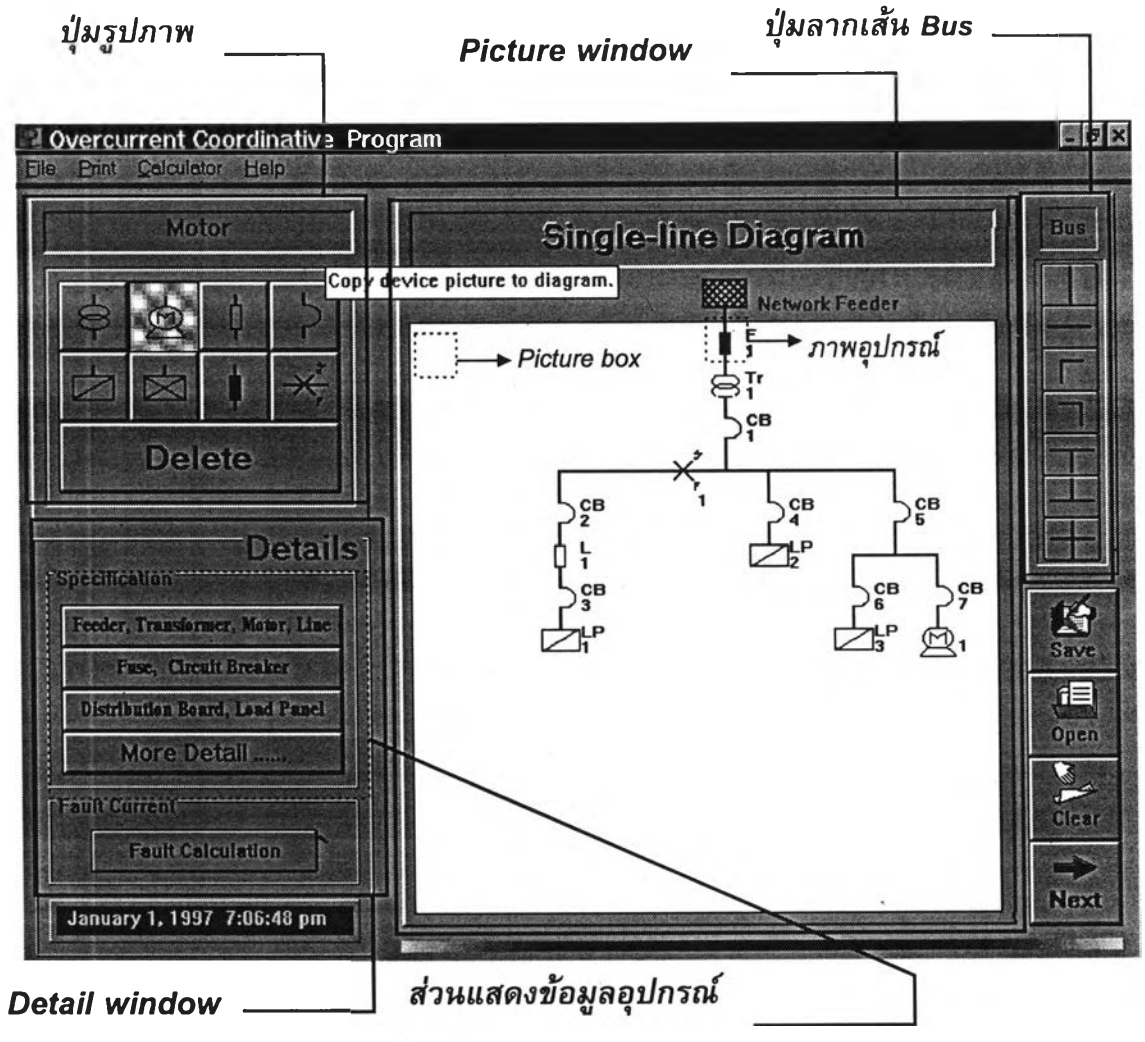


รูปที่ 7.2 วินโดว์แนะนำชื่อโปรแกรม (Introductory window)

หลังจากกดปุ่ม OK บน Introductory window แล้วจึงจะเริ่มเข้าสู่การใช้งานโปรแกรมในส่วนของการสร้างไดอะแกรมบน Single-line-diagram window ดังแสดงด้วยรูปที่ 7.3 ซึ่งถูกแบ่งเป็น 4 ส่วนย่อย คือ

- ส่วนของการสร้างรูปไดอะแกรม (Diagram creation)
- ส่วนของการให้และแสดงข้อมูลอุปกรณ์ (Equipment data)

- ส่วนคำนวณอิมพีแดนซ์ของอุปกรณ์ (Impedance calculation)
- ส่วนของแถบเมนู (Pop-up menu)



รูปที่ 7.3 Single-line-diagram window

โดยในแต่ละส่วนจะมีรายละเอียด ดังนี้

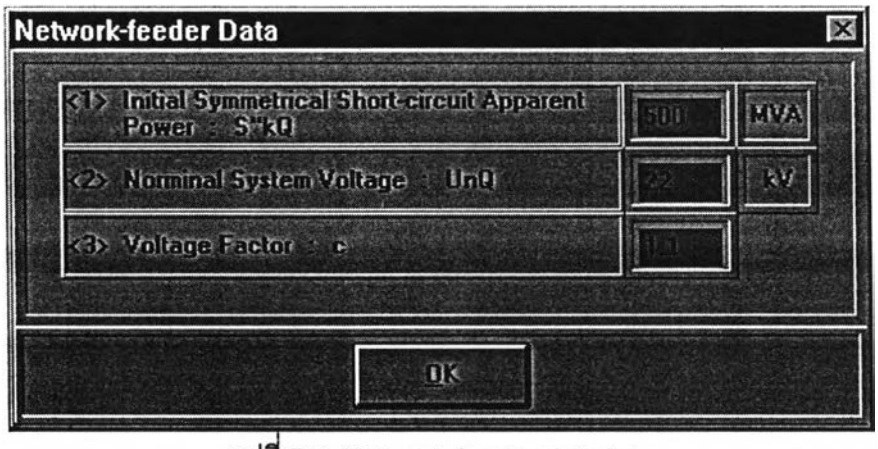
1.1) ส่วนของการสร้างรูปไดอะแกรม

อุปกรณ์ทั้งหมดที่จะปรากฏบนไดอะแกรมจะมีได้ 8 ชนิด ซึ่งแบ่งตามลักษณะการเขียนโปรแกรมเป็น 3 ประเภทดังนี้

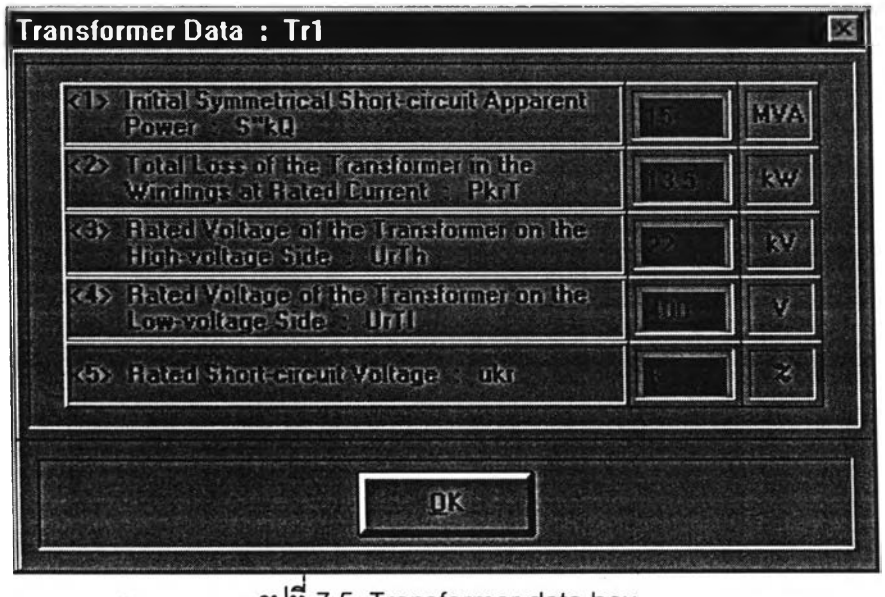
1. อุปกรณ์หลัก มี 4 ชนิด คือ **แหล่งจ่ายไฟฟ้า** (Network feeder : Q), **หม้อแปลงกำลัง** (Transformer : Tr), **มอเตอร์** (Motor : M) และ **สายส่ง** (Line : L)

ลักษณะพิเศษ คือ

- มี Data box เพื่อให้ผู้ใช้ใส่ข้อมูลของอุปกรณ์ ดังที่แสดงด้วยรูปที่ 7.4 - 7.7
- มีการคิดอิมพีแดนซ์
- มีการแสดงข้อมูลบน Detail window โดยอัตโนมัติ



รูปที่ 7.4 Network-feeder data box

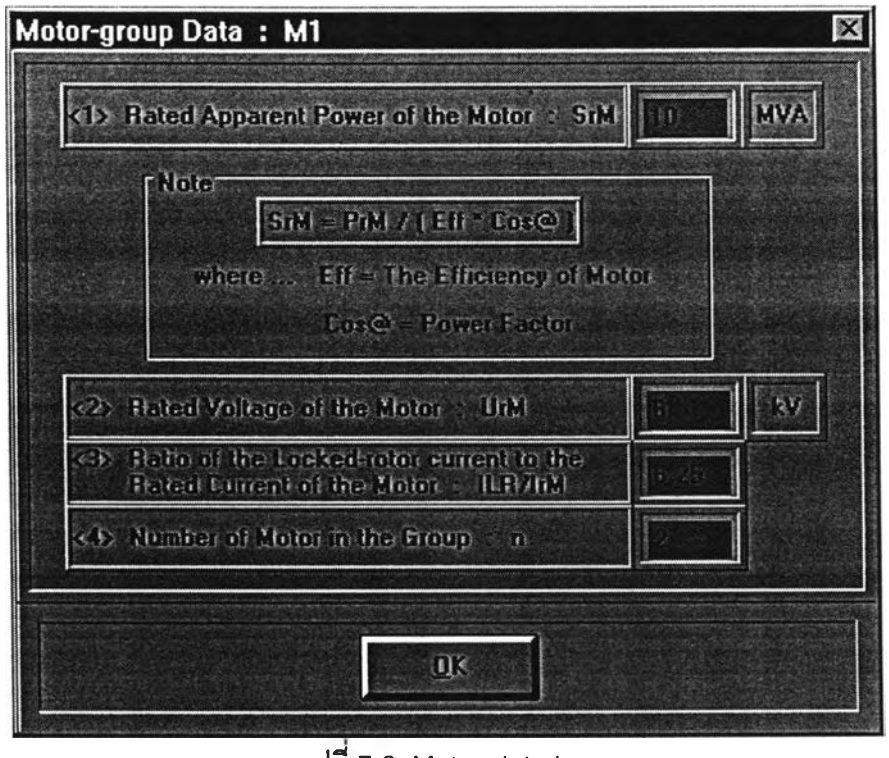


รูปที่ 7.5 Transformer data box

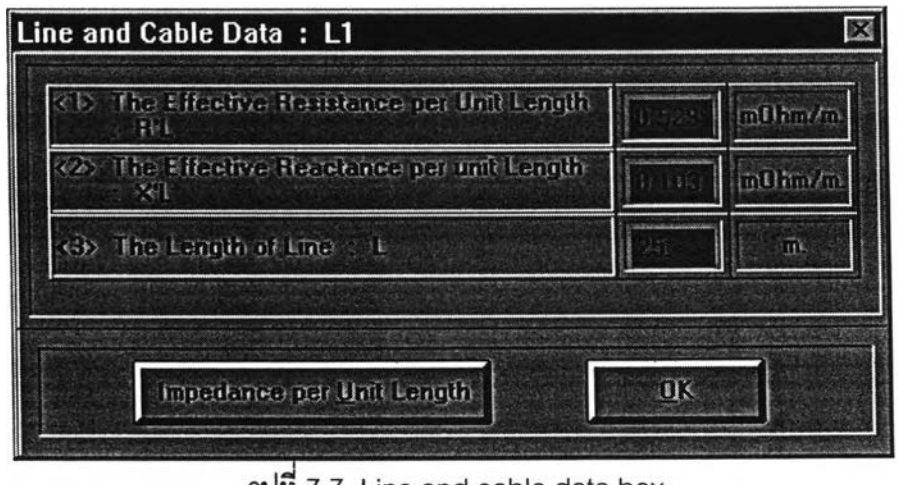
2. <u>อุปกรณ์ตัดวงจร</u> มี 2 ชนิด คือ **ฟิวส์** (Fuse : F) และ **เซอร์กิตเบรคเกอร์** (Circuit breaker : CB)

ลักษณะพิเศษ คือ

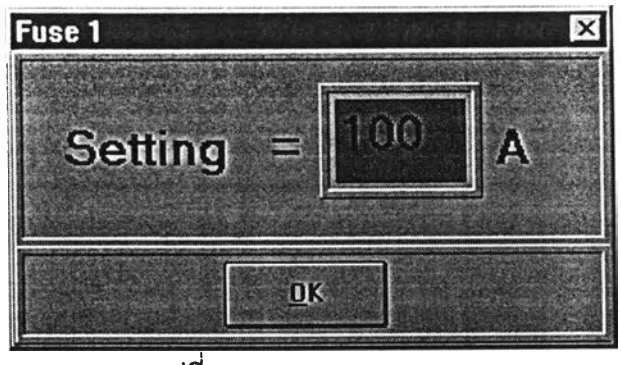
- มี Setting box เพื่อให้ผู้ใช้ใส่ขนาดพิกัดการทำงานของอุปกรณ์ ดังที่แสดงด้วยรูปที่ 7.8 - 7.9



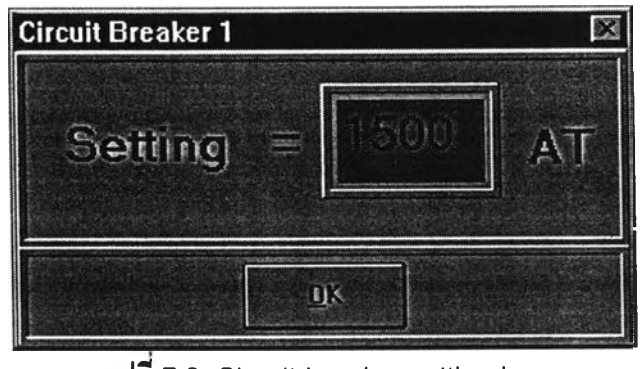
รูปที่ 7.6 Motor data box



รูปที่ 7.7 Line and cable data box



รูปที่ 7.8 Fuse setting box



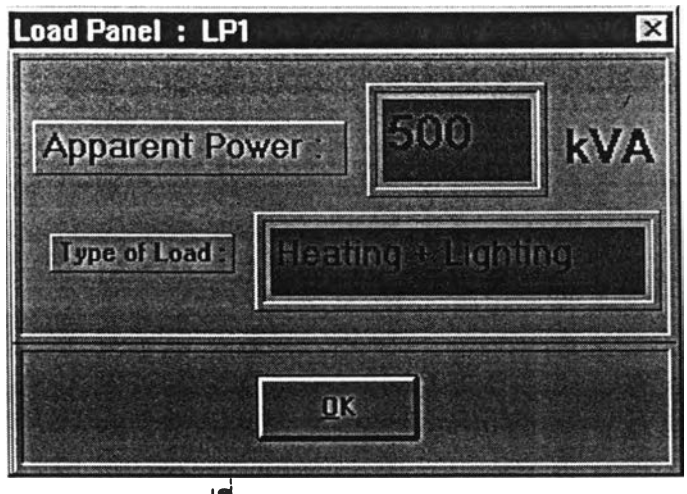
รูปที่ 7.9 Circuit-breaker setting box

3. <u>อุปกรณ์เสริม</u> **มี** 2 **ชนิด คือ แผงจ่ายโหลด** (Load panel : LP) **และ ตู้จ่ายไฟ** (Distribution board : DB)

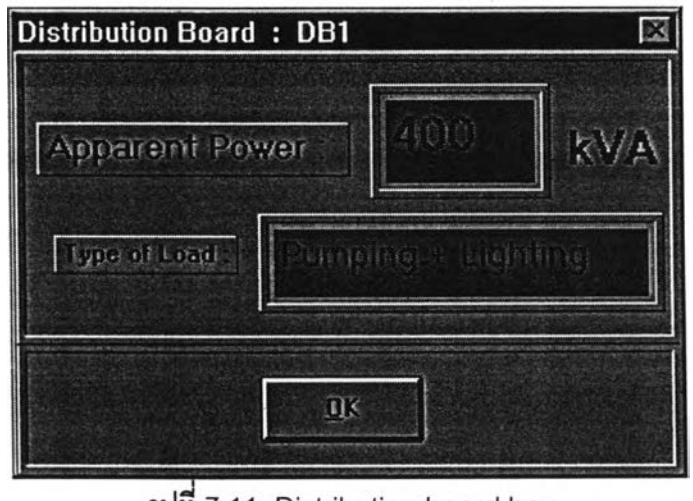
ลักษณะพิเศษ คือ

- **มี** Load box **เพื่อให้ผู้ใช้ใส่ขนาด และประเภทของโหลด ดังที่แสดงด้วยรูปที่** 7.10 - 7.11

แต่ละอุปกรณ์จะมีปุ่มรูปภาพให้ผู้ใช้กดในการถ่ายลง Picture box **เพื่อสร้างไดอะแกรม บน** Picture window **ยกเว้นสัญลักษณ์ของ** Network feeder **ซึ่งมีจุดเดียวบนไดอะแกรม จึงถูกกำหนดให้อยู่บน** Picture window **แล้ว ดังนั้นจึงไม่มีปุ่มรูปภาพของ** Network feeder **และนอกจากปุ่มรูปภาพของอุปกรณ์ทั้ง** 7 **ตัวแล้ว ยังมีปุ่มสัญลักษณ์ของ** "การกำหนดจุด Fault" **ด้วย**



รูปที่ 7.10 Load-panel box



รูปที่ 7.11 Distribution-board box

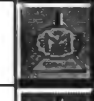
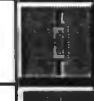
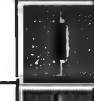
เมื่อลงภาพอุปกรณ์บน Picture window แล้ว ภาพของอุปกรณ์บางตัวจะมี อักษรย่อบอกชนิดของอุปกรณ์ และหมายเลขกำกับไว้ด้วยบน Picture box ซึ่งสามารถสรุปได้ดังตารางที่ 7.1 และ 7.2

ตารางที่ 7.1 สรุปรายละเอียดของปุ่มรูปภาพ และภาพอุปกรณ์บน Picture window

	Tr	M	L	F	CB	LP	DB	Fault
- อุปกรณ์ที่ต้องมีปุ่มรูปภาพ	/	/	/	/	/	/	/	/
- อุปกรณ์ที่ต้องมีอักษรย่อบอกชนิดของอุปกรณ์	/		/	/	/	/	/	/
- อุปกรณ์ที่มีหมายเลขกำกับ	/	/	/	/	/	/	/	/

จากตารางสรุปที่ 7.1 จึงนำไปสู่การพัฒนาโปรแกรม ซึ่งผลที่ได้สามารถแสดงให้เห็นได้จากตารางที่ 7.2

ตารางที่ 7.2 เปรียบเทียบระหว่างปุ่มรูปภาพ และภาพของอุปกรณ์บน Picture window

อุปกรณ์	ปุ่มรูปภาพ	ภาพของอุปกรณ์
1. Transformer		
2. Motor		
3. Line		
4. Fuse		
5. Circuit breaker		
6. Load panel		
7. Distribution board		
8. Fault point		

** หมายเลข 1 บนภาพอุปกรณ์ และจุด Fault เป็นหมายเลขสมมุติเพื่อแสดงให้เห็นลำดับการใช้ปุ่มนั้นเป็นลำดับแรก*

ในการใช้งานผู้ใช้ต้องลากเมาส์ (Mouse) ไปคลิ๊กที่ปุ่มรูปภาพของอุปกรณ์ที่ต้องการ (เมื่อคลิ๊กแล้ว ปุ่มกดที่ถูกคลิ๊กจะมีลักษณะแสดงให้เห็นว่าถูกกดลงไป ดังเช่นปุ่มกดของมอเตอร์ ที่ปรากฏในรูปที่ 7.3) จากนั้นจึงทำการเลื่อนเมาส์ เพื่อถ่าย (Copy) รูปนั้นลงในกล่องรูปภาพ (Picture box) ซึ่งอยู่ภายใน Picture window ในการ Copy รูปลงในกล่องรูปภาพนั้น ผู้ใช้ก็เพียงคลิ๊กที่ด้านซ้ายของเมาส์อีกครั้งลงในกล่องรูปภาพที่อยู่ในตำแหน่งที่ต้องการ ซึ่งในการกดปุ่มรูปภาพ 1 ครั้ง ผู้ใช้โปรแกรมจะสร้างภาพได้อย่างไม่จำกัดจำนวน (สังเกตได้จากปุ่มที่ถูกกดจะยังไม่คืนสภาพแม้ว่าจะทำการ Copy หลายรูปแล้วก็ตาม)

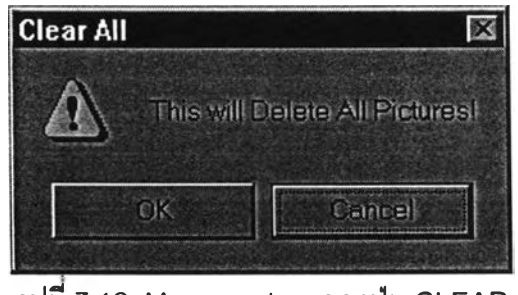
นอกจากนั้น ยังมีปุ่มกดอีก 3 ปุ่มเพื่อช่วยในการสร้างไดอะแกรม คือปุ่มลากเส้น Bus, ปุ่ม Delete และปุ่ม Clear โดยแต่ละปุ่มที่กล่าวมานั้น จะมีลักษณะดังต่อไปนี้

ในการสร้างเส้นบัสที่ทำการเชื่อมต่อระหว่างอุปกรณ์ทำได้โดยการคลิ๊กที่ในส่วนของเส้นบัสที่ต้องการบนชุดของกลุ่มเส้นบัสที่จัดไว้ให้ ดังที่เห็นในรูปที่ 7.3 จากนั้นจึงทำการลากเส้นบัสบนไดอะแกรม ซึ่งจะมีลักษณะการใช้งานเหมือนปุ่มรูปภาพของอุปกรณ์

ในการแก้ไขไดอะแกรมที่สร้างขึ้น โปรแกรมจะมี 2 ปุ่มให้ผู้ใช้โปรแกรมเลือก คือ

- ปุ่ม DELETE : ใช้ในการลบภาพของอุปกรณ์หรือส่วนของเส้นบัสที่ไม่ต้องการ ซึ่งทำได้โดยการเลื่อนเมาส์ไปคลิ๊กที่ปุ่ม DELETE ก่อนแล้วจึงไปคลิ๊กลงบนรูปภาพหรือแนวเส้นบัสที่ต้องการลบ โดยปุ่ม DELETE นี้จะทำการลบได้หลาย 1 รูปต่อการกด 1 ครั้ง

- ปุ่ม CLEAR : ใช้ในการลบภาพของอุปกรณ์และเส้นบัสทั้งหมดบนไดอะแกรม เมื่อผู้ใช้กดปุ่มนี้แล้วโปรแกรมจะยังไม่ทำการลบรูปภาพของอุปกรณ์และเส้นบัส ตราบจนกระทั่งผู้ใช้กดปุ่ม OK ภายใน Message box ที่ปรากฏออกมาหลังจากกดปุ่ม CLEAR โดยภายใน Message box จะมีการย้ำผู้ใช้โปรแกรมเพื่อให้แน่ใจว่าต้องการที่จะลบข้อมูลทั้งหมด ซึ่ง Message box นี้มีลักษณะดังรูปที่ 7.12



รูปที่ 7.12 Message box ของปุ่ม CLEAR

1.2) **ส่วนของการให้และแสดงข้อมูลอุปกรณ์**

หลังจากที่ผู้ใช้สร้างไดอะแกรมเสร็จเรียบร้อยแล้ว ผู้ใช้ต้องลงข้อมูลของอุปกรณ์โดยเริ่มจากการกดปุ่มที่มีชื่อของอุปกรณ์ปุ่มใดก็ได้ของส่วนแสดงข้อมูลอุปกรณ์บน Single-line-diagram window ในรูปที่ 7.3 หลังจากนั้นโปรแกรมจะจัดให้มีวินโดว์ข้อมูลมาให้ผู้ใช้กรอกรายละเอียดของอุปกรณ์แต่ละตัว ซึ่งวินโดว์ข้อมูลได้แบ่งเป็น 3 แบบตามประเภทของอุปกรณ์ คือ

1) Data box ของแหล่งจ่ายไฟฟ้า, หม้อแปลง, มอเตอร์ และสายส่ง ดังรูปที่ 7.4 - 7.7

2) Setting box ของฟิวส์ และเซอร์กิตเบรคเกอร์ ดังรูปที่ 7.8 - 7.9

3) Load box ของแผงจ่ายโหลด และตู้จ่ายไฟ ดังรูปที่ 7.10 - 7.11

หลังจากที่ผู้ใช้ใส่ข้อมูลในแต่ละวินโดว์ข้อมูลเสร็จก็ให้กดปุ่ม OK ในวินโดว์นั้นเพื่อให้โปรแกรมแสดงวินโดว์ข้อมูลในลำดับถัดไป ซึ่งจะเรียงลำดับตามรูปที่ 7.4 - 7.11 และสำหรับ

อุปกรณ์ที่มีอยู่หลายตัวในไดอะแกรม ทางโปรแกรมก็จะจัดให้มีวินโดว์ของอุปกรณ์นั้นๆ แสดงออกมาเท่ากับจำนวนอุปกรณ์ เช่น ถ้ามีกลุ่มของมอเตอร์อยู่ 3 กลุ่มในไดอะแกรม ผู้ใช้ก็จะต้องเติมรายละเอียดของมอเตอร์ทั้ง 3 กลุ่ม ผ่านทาง Data box ของมอเตอร์ทั้ง 3 วินโดว์ตามลำดับของมอเตอร์ M1 - M3

นอกจากการใส่รายละเอียดของอุปกรณ์ตามปกติดังที่กล่าวไปแล้วนั้น อุปกรณ์บางตัวจะมีลักษณะบางอย่างเพิ่มเข้ามา เพื่อให้การรับข้อมูลของโปรแกรมสมบูรณ์ยิ่งขึ้น ซึ่งได้แก่ การลงข้อมูลของแหล่งจ่ายไฟฟ้า คือหลังจากกดปุ่ม OK ใน Data box ของแหล่งจ่ายแล้ว โปรแกรมจะมี Message box แสดงออกมาถามผู้ใช้ว่าอุปกรณ์นี้ได้มีการติดตั้งในด้านแรงดันสูงของหม้อแปลงหรือไม่ดังรูปที่ 7.13

รูปที่ 7.13 Message box ถามลักษณะการติดตั้งของแหล่งจ่ายไฟฟ้า

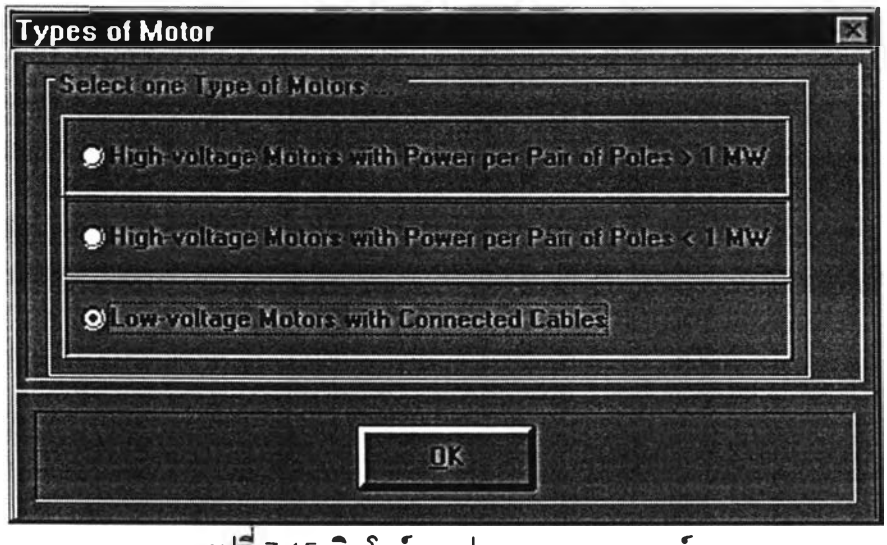
ซึ่งถ้าเป็นการติดตั้งในด้านแรงดันสูงจริง โปรแกรมจะจัดให้มีวินโดว์ขึ้นมาถามระดับแรงดันพิกัดทั้งด้านแรงดันสูงและต่ำของหม้อแปลงที่ต่ออยู่กับอุปกรณ์ตัวนั้น ดังรูปที่ 7.14 เพื่อนำไปคำนวณหาอัตราส่วนรอบ (Turn ratio) ของหม้อแปลง ซึ่งจะนำไปใช้ในการย้าย (Transfer) ค่าอิมพีแดนซ์อุปกรณ์ในด้านแรงดันสูงมาคำนวณในด้านแรงดันต่ำต่อไป

รูปที่ 7.14 วินโดว์ถามแรงดันพิกัดของหม้อแปลงเพื่อนำไปคำนวณหาอัตราส่วนรอบ (Turn ratio)

ในกรณีของการลงข้อมูลให้กับมอเตอร์ ซึ่งโดยปกติจะถูกแบ่งออกเป็น 3 ประเภท ได้แก่

1. มอเตอร์แรงดันสูงที่มีกำลังต่อคู่ของขั้ว $\geq$1 MW
2. มอเตอร์แรงดันสูงที่มีกำลังต่อคู่ของขั้ว <1 MW
3. กลุ่มของมอเตอร์แรงดันต่ำที่ต่อถึงกัน

เมื่อผู้ใช้ให้รายละเอียดของมอเตอร์ในแต่ละกลุ่มเสร็จ ผู้ใช้โปรแกรมจะต้องเลือกประเภทของมอเตอร์ที่ใช้ด้วยว่าเป็นประเภทไหนใน 3 ประเภทนี้ เพื่อคำนวณค่าอิมพีแดนซ์ในแต่ละกรณี โดยวินโดว์ที่ให้ผู้ใช้ระบุประเภทของมอเตอร์นี้จะเป็นดังรูปที่ 7.15



รูปที่ 7.15 วินโดว์ระบุประเภทของมอเตอร์

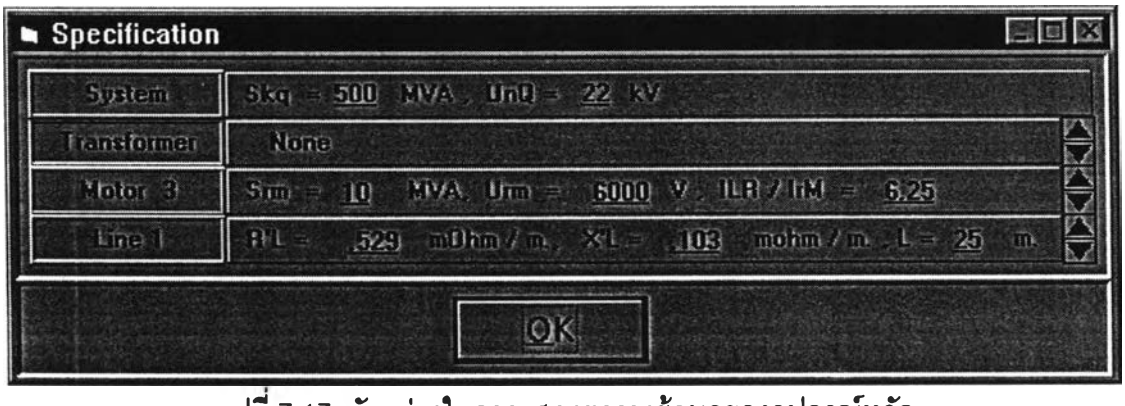
หลังจากที่ผู้ใช้ลงรายละเอียดอุปกรณ์ทุกตัวในไดอะแกรมเสร็จเรียบร้อยแล้ว ผู้ใช้สามารถลงรายละเอียดเพิ่มเติมได้ โดยกดปุ่ม More Detail ในส่วนแสดงข้อมูลอุปกรณ์ ซึ่งรายละเอียดที่กล่าวถึงนี้อาจจะเป็น ชื่อของบริษัทผู้ผลิต, ชื่อของรุ่นที่ใช้, วันที่ในการสั่งซื้อ ฯลฯ เพื่อเก็บไว้เป็นเอกสารอ้างอิง และสามารถพิมพ์ออกมาเก็บไว้ใช้ต่อไปได้

ที่กล่าวมาแล้วนั้นเป็นการลงข้อมูลให้กับอุปกรณ์ในไดอะแกรม แต่ในบางกรณีผู้ใช้อาจต้องการแก้ไขข้อมูลบางตัว ซึ่งสามารถทำได้โดยดับเบิลคลิ๊กที่ภาพของอุปกรณ์ที่ต้องการแก้ไขบนไดอะแกรม โดยโปรแกรมจะแสดงวินโดว์ข้อมูลของอุปกรณ์นั้นๆให้ผู้ใช้ได้แก้ไข ซึ่งข้อมูลใหม่นี้จะไปปรับ (Update) ค่าตัวแปร และส่วนแสดงข้อมูลอุปกรณ์ใน Detail window ด้วย

สำหรับการตรวจดูข้อมูลของอุปกรณ์ในระบบไฟฟ้า ผู้ใช้สามารถดูได้จากส่วนแสดงข้อมูลอุปกรณ์ภายใน Detail window ซึ่งโปรแกรมจะจัดการแสดงผลตามประเภทของอุปกรณ์ ซึ่งขึ้นอยู่กับปุ่มกดที่มีชื่อของอุปกรณ์ที่ผู้ใช้เลือกดังรูปที่ 7.16

รูปที่ 7.16 ปุ่มกดแสดงข้อมูลของอุปกรณ์

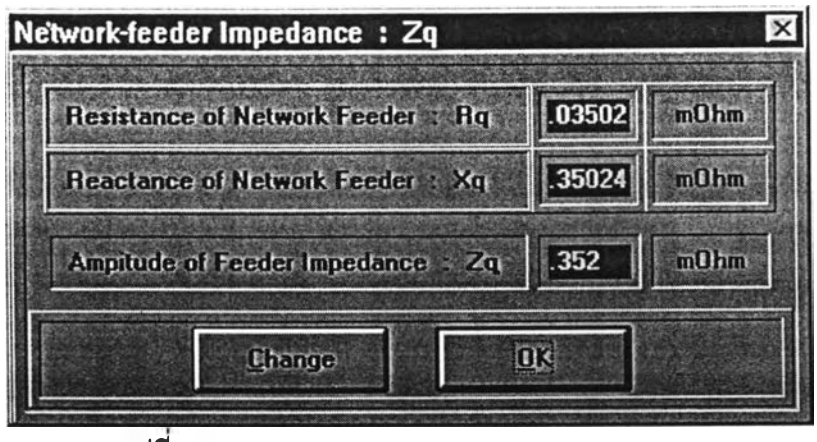
ในกรณีที่ผู้ใช้กดปุ่มแสดงข้อมูลอุปกรณ์หลัก ซึ่งประกอบด้วย Network feeder, Transformer, Motor และ Line โปรแกรมจะจัดให้มีตารางแสดงข้อมูลของอุปกรณ์หลักทั้ง 4 ประเภทนี้ ดังตัวอย่างรูปที่ 7.17 ซึ่งมีมอเตอร์อยู่ 3 กลุ่มในระบบไฟฟ้า ตารางแสดงข้อมูลดังที่กล่าวไปแล้วจะมีลูกศรกดขึ้น-ลง เพื่อให้ผู้ใช้สามารถเลื่อนดูข้อมูลของกลุ่มมอเตอร์ทั้ง 3 กลุ่มได้ แต่ในกรณีที่ไม่มีอุปกรณ์บางประเภทในระบบ โปรแกรมก็จะบอกผู้ใช้ว่าไม่มีอุปกรณ์นั้นๆ ดังตัวอย่างของหม้อแปลงในวินโดว์แสดงข้อมูล



รูปที่ 7.17 ตัวอย่างในการแสดงตารางข้อมูลของอุปกรณ์หลัก

1.3) **ส่วนคำนวณอิมพีแดนซ์ของอุปกรณ์**

การหาค่าอิมพีแดนซ์จะคิดเฉพาะอุปกรณ์ที่เป็น แหล่งจ่ายไฟฟ้า, หม้อแปลง, มอเตอร์, และสายไฟ โดยอุปกรณ์แต่ละตัวจะมีตัวแปรที่ต้องการ คือ ความต้านทาน (Resistance : R), รีแอกแตนซ์ (Reactance : X) และอิมพีแดนซ์ (Impedance : Z) $\left[Z = \sqrt{R^2 + X^2}\right]$ ซึ่งสามารถรู้ค่า R, X และ Z ของอุปกรณ์ได้จาก Impedance box ซึ่งปรากฏออกมาหลังจากกดปุ่ม OK ใน Data box ของอุปกรณ์แต่ละตัว โดยดูจากตัวอย่างรูปที่ 7.18 ซึ่งเป็น Impedance box ของ Network feeder โดยที่ค่าตัวแปรที่เห็นอาจเปลี่ยนแปลงได้ ถ้าผู้ใช้กดปุ่ม Change เพื่อเข้าไปแก้ไขข้อมูลของอุปกรณ์ใน Data box



รูปที่ 7.18 Impedance box ของ Network feeder

ค่าอิมพีแดนซ์ของอุปกรณ์แต่ละตัวมีหลักการ ดังนี้

1) **แหล่งจ่ายไฟฟ้า**

$$Z_S = \frac{cU_{nQ}^2}{S''_{kQ}} \frac{1}{t_r^2}$$ ... ตัวแปร คือ U_{nQ}, t_r และ S''_{kQ}

$X_S = 0.9950\ Z_S$

$R_S = 0.0995\ Z_S$

โดยที่ ค่าตัวประกอบแรงดัน c หาได้จากตารางที่ 6.1 ในบทที่ 6

2) **หม้อแปลง**

$$Z_T = \frac{u_{kT}}{100} \frac{U_{rT}^2}{S_{rT}}$$ ... ตัวแปร คือ u_{kr}, U_{rT}, S_{rT}

$$R_T = \frac{P_{krT}}{S^2_{rT}} \frac{U_{rT}^2}{}$$ ... ตัวแปร คือ P_{krT}

$$X_T = \sqrt{Z_T^2 - R_T^2}$$

3) **สายไฟ**

$R_L = R'_L * L$

$X_L = X'_L * L$ ... ตัวแปร คือ R'_L, X'_L, L

$$Z_L = \sqrt{R_L^2 + X_L^2}$$

นอกจากนั้นยังมีตาราง R_L และ X_L ที่ขนาดสายต่างๆให้ผู้ใช้เลือกด้วย โดยกดปุ่ม Impedance per Unit Length ใน Data box ของอุปกรณ์ Line

4) **มอเตอร์**

$$Z_M = \frac{1}{n} * \frac{1}{I_{LR} / I_{rM}} * \frac{U_{rM}^2}{S_{rM}}$$ ... ตัวแปร คือ I_{LR}/I_{rM}, U_{rM}, S_{rM}, n

ส่วนค่า R_M และ X_M นั้นให้เลือกประเภทของมอเตอร์ที่ใช้ก่อน ซึ่งถูกแบ่งเป็น 3 ประเภท ดังนี้

1. สำหรับมอเตอร์แรงดันสูงที่มีกำลังต่อคู่ของขั้ว $\geq$1 MW : $X_M = 0.995\ Z_M$, $R_M = 0.09950 Z_M$
2. สำหรับมอเตอร์แรงดันสูงที่มีกำลังต่อคู่ของขั้ว $<$1 MW : $X_M = 0.989\ Z_M$, $R_M = 0.14835 Z_M$
3. สำหรับกลุ่มของมอเตอร์แรงดันต่ำที่ต่อถึงกัน : $X_M = 0.922\ Z_M$, $R_M = 0.38724 Z_M$

1.4) **ส่วนของแถบเมนู**

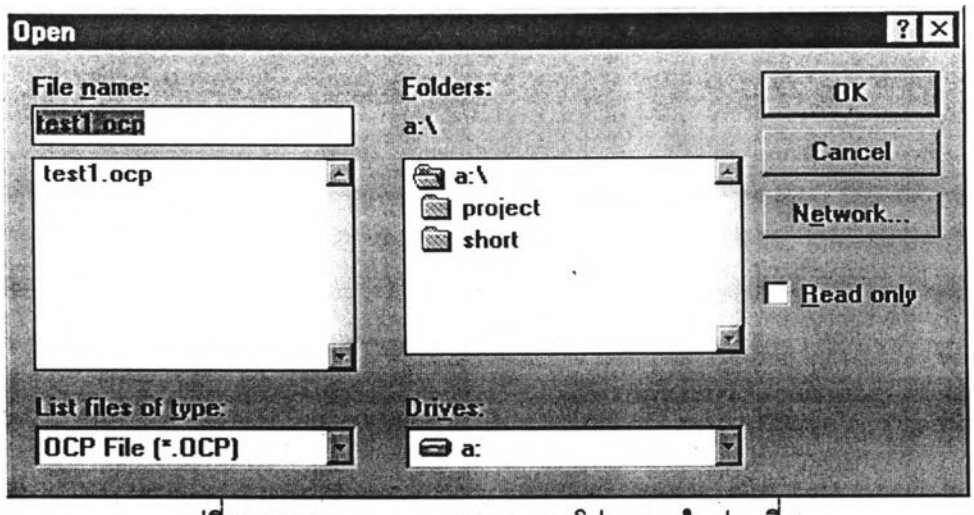
ภายใน Single-line-diagram window จะมีแถบเมนู (Menu bar) ดังจะเห็นได้ในตำแหน่งมุมบนซ้ายของรูปที่ 7.3 ผู้ใช้โปรแกรมสามารถเลือกใช้ตัวเลือกของคำสั่งต่างๆ ซึ่งประกอบไปด้วย

1.4.1) File menu : (หรือกด ALT-F) ใช้ในการเริ่มต้นใช้งาน (New), เปิดแฟ้มงานเก่า (Open), จัดเก็บข้อมูล (Save & Save As) และหยุดโปรแกรม (Exit) ตามลักษณะของตัวเลือก (Option) ที่ผู้ใช้โปรแกรมคลิ๊กซึ่งมีลักษณะเป็นแบบ Pull-Down โดยแต่ละตัวเลือกจะมีรายละเอียดดังนี้

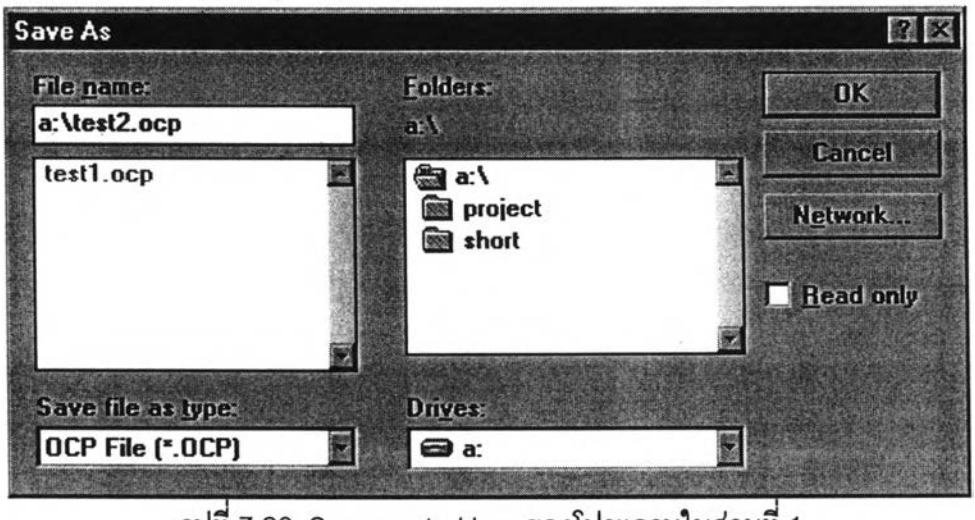
- New option : ใช้ในการเริ่มสร้างไดอะแกรมเป็นแฟ้มเก็บข้อมูล (File) ใหม่ ซึ่งเมื่อผู้ใช้โปรแกรมคลิ๊กลงบนตัวเลือกนี้ งานเก่าจะถูกลบทั้งหมด และพร้อมที่จะให้ผู้ใช้โปรแกรมเริ่มสร้างไดอะแกรม และลงข้อมูลใหม่

- Open option : ใช้ในการเปิดไฟล์ที่มีข้อมูลเดิมอยู่แล้วจากจานบันทึกข้อมูล (Diskette) หรือภายในฮาร์ดดิสก์ (Hard disk) ของเครื่องคอมพิวเตอร์ ซึ่งตัวอย่างการใช้งานในการเปิดไฟล์ที่ชื่อ test1.ocp ที่เก็บไว้ในจานบันทึกข้อมูลจากตัวขับ A เป็นดังรูปที่ 7.19 ซึ่งข้อมูลทั้งหมดในไฟล์นี้จะปรากฏออกมาในโปรแกรมเมื่อผู้ใช้กดปุ่ม OK ที่อยู่ภายใน Open control box

- Save As option : ใช้ในการจัดเก็บข้อมูลของไดอะแกรม และรายละเอียดทั้งหมดที่ผู้ใช้โปรแกรมสร้างขึ้น โดยที่ผู้ใช้โปรแกรมสามารถตั้งชื่อหรือเลือกไฟล์ และตัวขับจานบันทึก (Disk drive) ของคอมพิวเตอร์ที่ต้องการจะจัดเก็บ โดยพิจารณาจากรูปที่ 7.20 ซึ่งเป็นการจัดเก็บงานลงใน ไฟล์ที่ชื่อ test2.ocp ภายในตัวขับ A



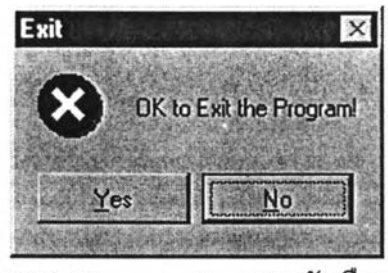
รูปที่ 7.19 Open control box ของโปรแกรมในส่วนที่ 1



รูปที่ 7.20 Save control box ของโปรแกรมในส่วนที่ 1

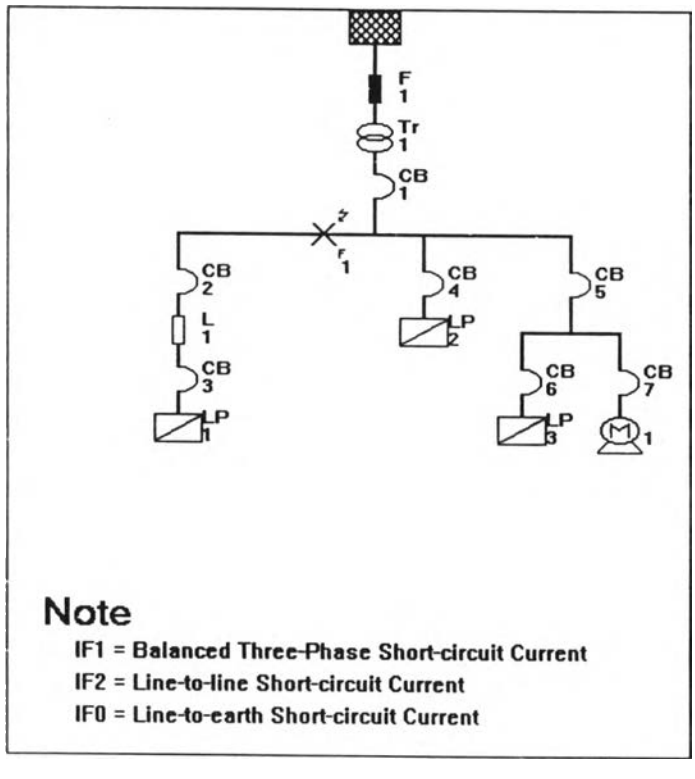
- Save option : เช่นเดียวกับ Save As option แต่ผู้ใช้ไม่จำเป็นต้องระบุชื่อและตำแหน่ง (Path) ของไฟล์

- Exit option : ใช้เมื่อผู้ใช้โปรแกรมต้องการออกจากโปรแกรม Overcurrent Coordinative Program โดยโปรแกรมจะไม่ให้ผู้ใช้โปรแกรมออกทันทีเมื่อกดตัวเลือก Exit แต่จะมี Message box ดังรูปที่ 7.21 มาถามย้ำความแน่ใจก่อนจะออกโปรแกรม และถ้าผู้ใช้โปรแกรมกดปุ่ม OK ใน Message box โปรแกรมนี้จึงจะถูกลบออกไปทันที



รูปที่ 7.21 Message box ของตัวเลือก Exit

1.4.2) Print menu : (หรือกด ALT-P) เพื่อพิมพ์ไดอะแกรม และรายละเอียดต่างๆ ที่ผู้ใช้โปรแกรมสร้างขึ้นผ่านทางเครื่องพิมพ์ (Printer) ที่เชื่อมอยู่ โดยโปรแกรมจะพิมพ์ข้อมูลทั้งหมดออกมาในรูปแบบที่ได้กำหนดไว้แล้ว ซึ่งตัวอย่างแสดงไว้ในรูปที่ 7.22



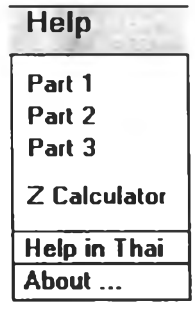
Feeder : SkQ = 500 MVA, UnQ = 22 KV
Tr 1 : Srt = 2000 KVA, PkrT = 20 KW, UrtH = 22 KV
UrtL = 400V, Ukr = 6 %
Motor 1 : SrM = 500 KVA, UrM = 400 V
I(LR) / I(rM) = 6.25
Line 1 : R'L = .74 mOhm/m, X'L = .107 mOhm/m
L = 40 m
LP 1 : 400 KVA
Type =Lighting + Heating
LP 2 : 500 KVA
Type =Lighting + Heating
LP 3 : 100 KVA
Type =Lighting + Pumping
Fuse 1 : Operating Ampere = 125
CB 1 : Ampere Trip = 3000
CB 2 : Ampere Trip = 800
CB 3 : Ampere Trip = 800
CB 4 : Ampere Trip = 1000
CB 5 : Ampere Trip = 1400
CB 6 : Ampere Trip = 200
CB 7 : Ampere Trip = 1200

Fault Point : 1
IF1 =42.65 kA
IF2 =49.81 kA
IF0 =49.25 kA

รูปที่ 7.22 ตัวอย่างของข้อมูลที่พิมพ์ออกมา

1.4.3) Help menu : (หรือกด ALT-H) ใช้เมื่อผู้ใช้โปรแกรมต้องการรู้ลักษณะและการใช้งานของโปรแกรมซึ่งถูกแบ่งหัวข้อดังรูปที่ 7.23 และภายหลังจากที่ผู้ใช้โปรแกรมคลิ๊กลงบนตัวเลือก Help จากนั้นจะมี Help window ซึ่งมีข้อความอธิบายรายละเอียดของโปรแกรมอย่างกระชับและ

เข้าใจง่ายปรากฏออกมาบนหน้าจอของคอมพิวเตอร์ โดยสามารถดูคำอธิบายได้ในภาคผนวก ก. ของวิทยานิพนธ์ฉบับนี้



รูปที่ 7.23 **หัวข้อในการใช้คำสั่ง** Help

7.4.2 **ส่วนคำนวณหาค่ากระแสลัดวงจร**

โปรแกรมในส่วนนี้มีวัตถุประสงค์เพื่อให้ผู้ใช้สามารถคำนวณค่ากระแสลัดวงจรที่จุดต่างๆ ในไดอะแกรมเส้นเดี่ยวที่สร้างขึ้นได้โดยสะดวก โดยใช้อุปกรณ์พิเศษมาช่วยให้ผู้ใช้สามารถคำนวณค่าผลรวมของอิมพีแดนซ์ ณ.จุดลัดวงจรต่างๆได้ง่ายขึ้นแทนการคำนวณแบบ Manual ซึ่งค่ากระแสลัดวงจรที่กล่าวถึงนี้ไม่ได้เจาะจงเฉพาะการลัดวงจรในกรณีแบบสามเฟสสมดุล (Balanced three-phase short circuit) เท่านั้น แต่ยังรวมถึงการลัดวงจรระหว่างสายโดยไม่เกี่ยวข้องกับพื้นดิน (Line-to-line short circuit without earth connection) และการลัดวงจรระหว่างสายกับพื้นดิน (Line-to-earth short circuit) โดยผลคำนวณที่ได้ของการลัดวงจรทั้งหมดนี้จะถูกจัดเก็บไว้ในไฟล์เดียวกับไดอะแกรมในส่วนแรกด้วย

7.4.2.1 **ลักษณะของโปรแกรมส่วนที่** 2

- ผู้ใช้สามารถเลือกคำนวณค่ากระแสลัดวงจรเฉพาะแบบสามเฟสสมดุลเพียงกรณีเดียวได้
- มีการแสดงวงจรอิมพีแดนซ์ของไดอะแกรมเส้นเดี่ยวพร้อมทั้งสามารถกำหนดตำแหน่งของแหล่งจ่ายไฟสมมูลที่จะเกิดขึ้นในกรณีที่เกิดการลัดวงจรที่จุดต่างๆ (โดยในที่นี้กำหนดให้มีจุดผิดพร่องได้ไม่เกิน 4 จุด บนไดอะแกรมเส้นเดี่ยว)
- มีอุปกรณ์พิเศษที่ชื่อ Complex Calculator ซึ่งถูกออกแบบเพื่อช่วยให้ผู้ใช้สามารถคำนวณหาผลรวมของอิมพีแดนซ์เมื่อเกิดการลัดวงจรที่จุดผิดพร่อง ณ.จุดต่างๆได้ง่ายขึ้น โดยกระบวนการคิดคำนวณนี้จะเป็นการจัดการกับเลขเชิงซ้อน (Z=R+jX)

- มีการเชื่อมต่อกับโปรแกรมในส่วนที่ 1 (โดยกดปุ่ม Back บน Short-circuit-calculation window ซึ่งจะนำไปสู่ Single-line-diagram window ของโปรแกรมในส่วนแรกที่ใช้ในการสร้างไดอะแกรม และรับข้อมูลของอุปกรณ์)

- มีคำอธิบายการใช้โปรแกรม (Help) เป็นภาษาอังกฤษทั้งในส่วนคำนวณหากระแสลัดวงจร และในส่วนของ Complex Calculator

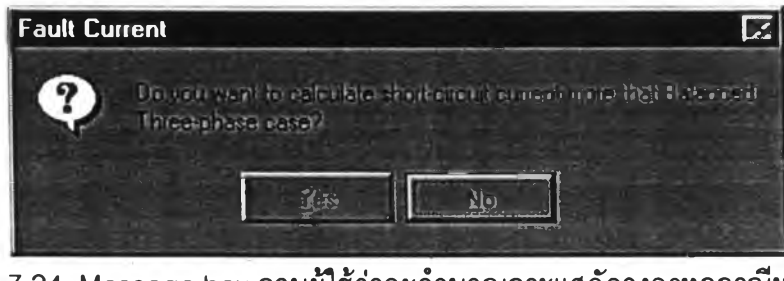
- ค่ากระแสลัดวงจรทุกกรณี และทุกจุดผิดพร่องจะถูกพิมพ์ออกมาพร้อมกับข้อมูลของโปรแกรมในส่วนแรก

7.4.2.2 รูปแบบของโปรแกรมส่วนที่ 2

ผู้ใช้สามารถเข้าสู่โปรแกรมในส่วนที่ 2 ได้โดยกดปุ่ม Fault Calculation ที่อยู่ใน Single-line-diagram window (รูปที่ 7.3) ของโปรแกรมส่วนแรก แต่ก่อนที่จะเข้าไปยังหน้าจอทำงานของโปรแกรมในส่วนที่สอง หน่วยสั่งงานของโปรแกรมจะแสดง Message box ดังรูปที่ 7.24 เพื่อถามผู้ใช้ว่าต้องการหาค่ากระแสลัดวงจรนอกเหนือจากแบบสามเฟสสมดุล (Balanced short-circuit current) หรือไม่ ซึ่งนอกจากกระแสลัดวงจรแบบนี้แล้ว ผู้ใช้สามารถคำนวณกระแสลัดวงจรได้อีก 2 ประเภท คือ

- กระแสลัดวงจรระหว่างสายโดยไม่เกี่ยวข้องกับพื้นดิน (Line-to-line short circuit current without earth connection)

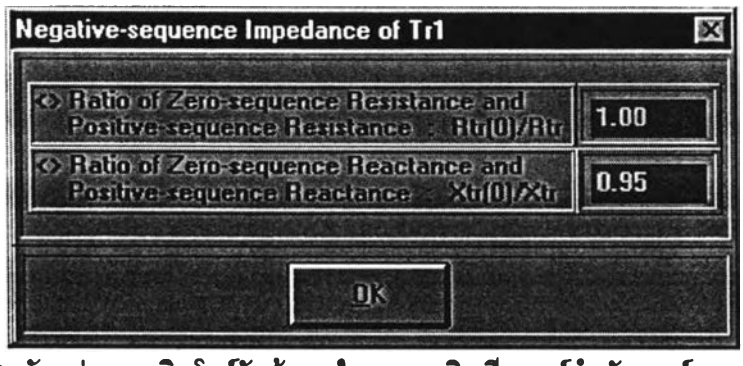
- กระแสลัดวงจรระหว่างสายกับพื้นดิน (Line-to-earth short circuit current)



รูปที่ 7.24 Message box ถามผู้ใช้ว่าจะคำนวณกระแสลัดวงจรทุกกรณีหรือไม่

ถ้าผู้ใช้ต้องการหาค่ากระแสลัดวงจรทั้ง 3 แบบ โปรแกรมจะจัดให้มีวินโดว์ข้อมูลมารับรายละเอียดเพิ่มเติมของอุปกรณ์หลักทั้ง 3 ตัว ได้แก่ หม้อแปลงกำลัง, มอเตอร์ และสายส่ง ซึ่งสิ่งที่ผู้ใช้โปรแกรมต้องเติมลงในวินโดว์เหล่านี้ คือค่าอัตราส่วนระหว่างอิมพีแดนซ์ลัดวงจรแบบลำดับศูนย์ (Zero-sequence impedance) และอิมพีแดนซ์ลัดวงจรแบบลำดับบวก (Positive-sequence

impedance) โดยวินโดว์รับข้อมูลเหล่านี้จะถูกแสดงออกมารับข้อมูลจากผู้ใช้เช่นเดียวกับวินโดว์รับข้อมูลในการหาอิมพีแดนซ์ลำดับบวกของโปรแกรมในส่วนแรก คือจะถูกแสดงแยกตามลำดับประเภทอุปกรณ์ และจะมีจำนวนเท่ากับอุปกรณ์หลักที่มีอยู่จริงบนไดอะแกรมเส้นเดี่ยว ซึ่งตัวอย่างวินโดว์รับข้อมูลในการหาอิมพีแดนซ์ลำดับศูนย์ของหม้อแปลงตัวแรก จะเป็นดังรูปที่ 7.25

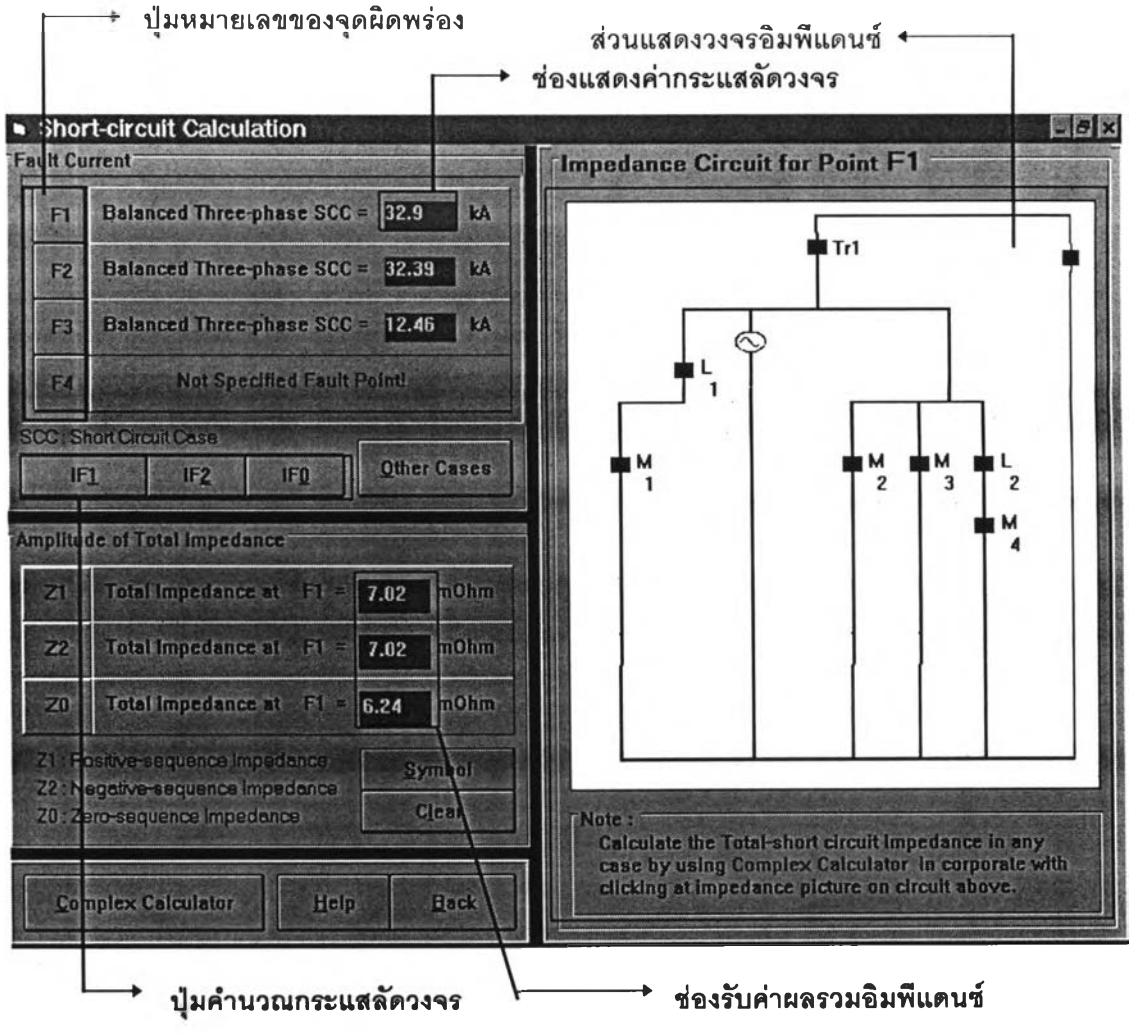


รูปที่ 7.25 ตัวอย่างของวินโดว์รับข้อมูลในการหาอิมพีแดนซ์ลำดับศูนย์ของหม้อแปลง

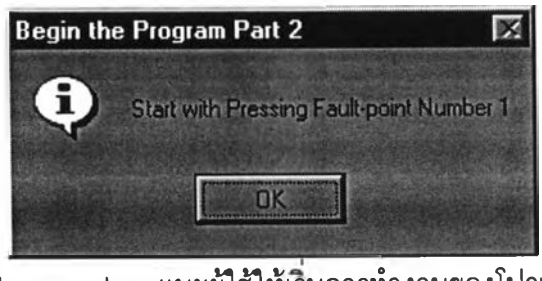
เมื่อลงอัตราส่วนระหว่างอิมพีแดนซ์ลำดับศูนย์ และอิมพีแดนซ์ลำดับบวกเสร็จแล้ว จึงจะเริ่มเข้าสู่หน้าจอการทำงานที่มีชื่อว่า Short-circuit-calculation window ดังแสดงด้วยรูปที่ 7.26 ซึ่งจะปรากฏออกมาพร้อมกับ Message box ดังรูปที่ 7.27 เพื่อแนะผู้ใช้ให้เริ่มการทำงานโดยกดปุ่มหมายเลขของจุดผิดพร่อง F1 ก่อน

ภายใน Short-current-calculation window นี้จะถูกแยกการทำงานได้เป็น 3 ส่วนหลัก คือ

1) **ส่วนของวงจรอิมพีแดนซ์** : แสดงวงจรที่เป็นผลมาจากไดอะแกรมเส้นเดี่ยวของระบบไฟฟ้าที่ผู้ใช้สร้างขึ้นในโปรแกรมส่วนแรก นอกจากนั้นก็จะมีการกำหนดแหล่งจ่ายไฟสมมูลที่ถูกสมมุติขึ้นมาจากการลัดวงจร ณ.จุดต่างๆบนวงจรตามที่ผู้ใช้กำหนดในไดอะแกรมเส้นเดี่ยวด้วย โดยตำแหน่งของแหล่งจ่ายไฟสมมูลนี้จะขึ้นอยู่กับปุ่มหมายเลขของจุดผิดพร่องที่ถูกกดในขณะนั้นๆ โดยพิจารณาจากตัวอย่างของวงจรอิมพีแดนซ์ที่อยู่ด้านขวาของหน้าจอรูปที่ 7.26 ซึ่งกำลังแสดงแหล่งจ่ายไฟสมมูลในกรณีของการลัดวงจรที่จุด F1



รูปที่ 7.26 Short-circuit-calculation window



รูปที่ 7.27 Message box แนะผู้ใช้ให้เริ่มการทำงานของโปรแกรมส่วนที่ 2 โดยการกดปุ่มหมายเลขของจุดผิดพร่อง F1

2) **ส่วนคำนวณกระแสลัดวงจร** : ผลคำนวณที่ได้ในช่องแสดงค่ากระแสลัดวงจรจะเป็นผลมาจากการคำนวณภายหลังการกดปุ่ม IF1, IF2 และ IF0 โดยที่

- ปุ่ม IF1 เป็นการคำนวณ กระแสลัดวงจรแบสมดุล 3 เฟสสมดุล (Balanced three-phase short circuit current)

- ปุ่ม IF2 เป็นการคำนวณ กระแสลัดวงจรระหว่างสายโดยไม่เกี่ยวข้องกับพื้นดิน (Line-to-line short circuit current without earth connection)

- ปุ่ม IF0 เป็นการคำนวณ กระแสลัดวงจรระหว่างสายกับพื้นดิน (Line-to-earth short circuit current)

แต่ในกรณีที่ผู้ใช้เลือกคำนวณเฉพาะค่ากระแสลัดวงจรแบบ 3 เฟสสมดุล โปรแกรมจะควบคุมให้ปุ่ม IF2 และ IF0 อยู่ในสภาพที่ไม่สามารถทำงานได้ (Disabled status) **ซึ่งการที่โปรแกรมสั่งให้ปุ่มกดบางปุ่มมีสภาพที่ไม่สามารถใช้งานได้เช่นนี้ จะเป็นเช่นเดียวกับปุ่มหมายเลขของจุดผิดพร่อง (F1-F4) เช่นในกรณีที่ผู้ใช้กำหนดจุดผิดพร่อง 3 จุดบนไดอะแกรมเส้นเดี่ยวของโปรแกรมในส่วนแรก ปุ่มหมายเลขของจุดผิดพร่อง F4 ก็จะไม่สามารถทำงานได้ ซึ่งโปรแกรมจะระบุข้อความด้วยว่าจุดผิดพร่องนี้ไม่ได้ถูกกำหนดบนไดอะแกรมเส้นเดี่ยว** ("Not Specified Fault Point!") **ดังในรูปที่** 7.26

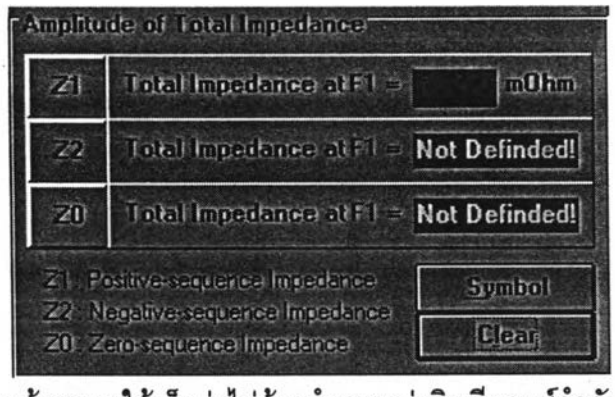
นอกจากนั้นก็ยังมีปุ่ม Other Cases **ที่ถูกใช้ในกรณีที่ผู้ใช้คำนวณกระแสลัดวงจรทุกกรณี โดยปุ่มนี้เมื่อถูกกดแล้วจะทำให้ประเภทของกระแสลัดวงจรพร้อมทั้งค่าที่คำนวณได้บนหน้าจอเปลี่ยนไปตามลำดับประเภทของการลัดวงจร แต่ในกรณีที่ผู้ใช้คำนวณกระแสลัดวงจรเฉพาะแบบสามเฟสสมดุล โปรแกรมจะควบคุมให้ปุ่มนี้ไม่สามารถใช้งานได้** (Disabled status)

3) **ส่วนคำนวณหาอิมพีแดนซ์ลัดวงจร** : ค่าอิมพีแดนซ์ที่กล่าวถึงนี้ **เป็นผลรวมของอิมพีแดนซ์ที่ผู้ใช้ทำการยุบวงจร ณ.จุดผิดพร่องในขณะที่กำลังวิเคราะห์ และเมื่อผู้ใช้คำนวณค่าอิมพีแดนซ์ทั้งแบบลำดับบวก, ลำดับลบ และลำดับศูนย์ (ในกรณีที่ผู้ใช้มีการคำนวณกระแสลัดวงจรทุกกรณี) หรือคำนวณเฉพาะอิมพีแดนซ์ลำดับบวกเสร็จแล้ว (ในกรณีที่คำนวณกระแสลัดวงจรเฉพาะแบบสามเฟสสมดุลเพียงแบบเดียว) จากนั้นผู้ใช้ก็สามารถสั่งให้โปรแกรมนำผลรวมอิมพีแดนซ์เหล่านี้ไปคำนวณหาค่ากระแสลัดวงจรในแต่ละกรณีได้โดยกดปุ่ม IF1, IF2 หรือ IF0 (ปุ่มที่ถูกกดให้คำนวณค่าไปแล้วจะมีการควบคุมให้สีของตัวหนังสือบนปุ่มคำสั่งเหล่านี้เปลี่ยนไปด้วย เพื่อให้รู้ว่าค่ากระแสลัดวงจรประเภทนั้นๆ ถูกสั่งให้คำนวณไปแล้ว แต่ปุ่มคำนวณทั้งสามนี้ จะกลับเป็นเช่นเดิมอีกครั้ง เมื่อผู้ใช้เริ่มคำนวณที่จุดผิดพร่องจุดใหม่นี้โดยการกดปุ่มหมายเลขของจุดผิดพร่องในลำดับถัดไป)**

เนื่องจากขนาดของอิมพีแดนซ์ลำดับบวกจะเท่ากับขนาดของอิมพีแดนซ์ลัดวงจรลำดับลบ ดังนั้น ผู้ใช้จึงคำนวณเฉพาะค่าอิมพีแดนซ์ลำดับบวก (Z1) **และอิมพีแดนซ์ลำดับศูนย์** (Z0) **เท่านั้น**

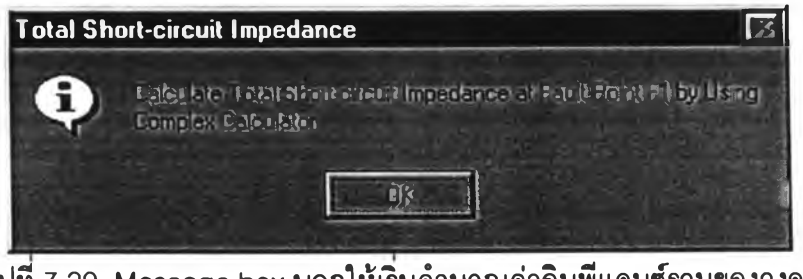
(ในกรณีที่คำนวณหากระแสลัดวงจรทุกกรณี) ส่วนค่าอิมพีแดนซ์ลำดับลบนั้นจะถูกกำหนดลงในช่องรับค่าพร้อมกันกับที่มีการถ่ายค่าลงในช่องรับค่าของอิมพีแดนซ์ลำดับบวกโดยอัตโนมัติ

ในกรณีที่ผู้ใช้ต้องการคำนวณกระแสลัดวงจรเฉพาะแบบสามเฟสสมดุล ซึ่งต้องการข้อมูลเพียงแค่ผลรวมของอิมพีแดนซ์ลำดับบวก (Z1) เท่านั้น ดังนั้นจึงไม่ต้องคำนวณผลรวมของอิมพีแดนซ์ลำดับลบ และลำดับศูนย์ ซึ่งโปรแกรมจะแสดงให้เห็นว่าข้อมูลทั้งสองนี้ไม่จำเป็นต้องใช้โดยมีแถบข้อความระบุให้เห็นในช่องรับค่าผลรวมของอิมพีแดนซ์ ดังรูปที่ 7.28 นอกจากนั้นก็ยังมีปุ่ม Clear เพื่อใช้ในกรณีที่ต้องการลบค่าผลรวมอิมพีแดนซ์ที่ลงไปแล้วในช่องรับค่า



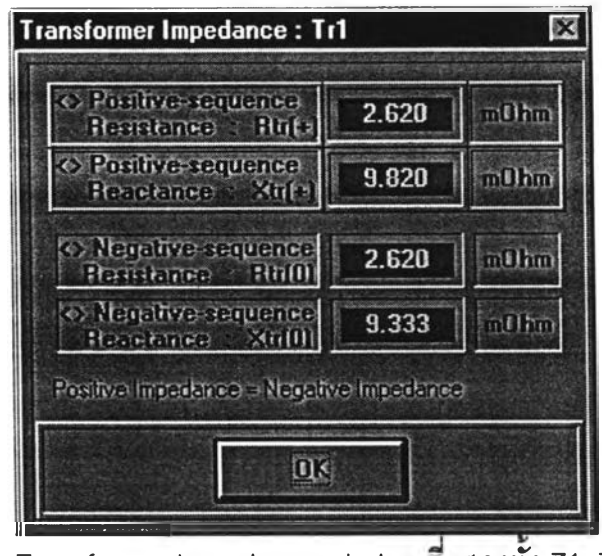
รูปที่ 7.28 การแสดงข้อความให้เห็นว่าไม่ต้องคำนวณค่าอิมพีแดนซ์ลำดับลบ และลำดับศูนย์ในการคำนวณค่ากระแสลัดวงจร

จากที่กล่าวไปแล้วว่าผู้ใช้จะเริ่มการทำงานของโปรแกรมในส่วนที่ 2 โดยการกดปุ่มหมายเลขของจุดผิดพร่อง F1 ซึ่งหลังจากนั้นจะมี Message box ดังรูปที่ 7.29 บอกให้ผู้ใช้เรียกอุปกรณ์พิเศษที่ชื่อ Complex Calculator เพื่อนำไปใช้ในการยุบรวมอิมพีแดนซ์ที่เกิดจากการลัดวงจรที่จุด F1 โดยพิจารณาร่วมกับวงจรอิมพีแดนซ์ที่แสดงให้เห็นทางด้านขวามือของหน้าจอ



รูปที่ 7.29 Message box บอกให้เริ่มคำนวณค่าอิมพีแดนซ์รวมของวงจรในกรณีที่เกิดจุดผิดพร่องที่จุด F1

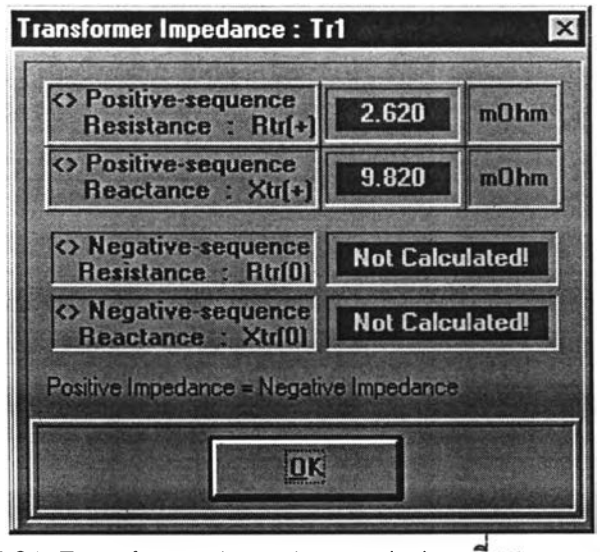
ในการยุบรวมค่าอิมพีแดนซ์จากวงจรนั้น นอกจากต้องใช้ Complex Calculator แล้วผู้ใช้จะต้องรู้ถึงค่าอิมพีแดนซ์ของอุปกรณ์แต่ละตัวบนไดอะแกรมด้วย ซึ่งผู้ใช้สามารถรู้ได้โดยดับเบิลคลิ๊กที่รูปสัญลักษณ์อิมพีแดนซ์บนวงจรแต่ละตัว จากนั้นจะมีการแสดงวินโดว์เพื่อบอกค่าอิมพีแดนซ์ของอุปกรณ์นั้นๆออกมา ดังตัวอย่างรูปที่ 7.30 ซึ่งเป็นของหม้อแปลงกำลังตัวที่ 1



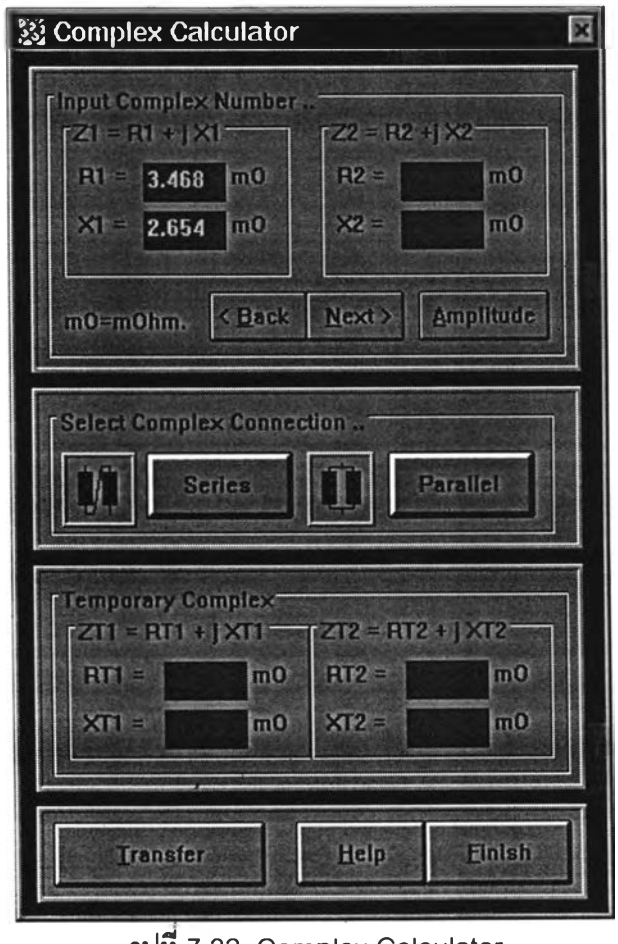
รูปที่ 7.30 Transformer-impedance window ที่แสดงทั้ง Z1, Z2 และ Z0

จากรูปที่ 7.30 จะเห็นว่ามีการแสดงค่าอิมพีแดนซ์ทั้ง Z1, Z2 และ Z0 ซึ่งเป็นวินโดว์แสดงค่าอิมพีแดนซ์กรณีที่ผู้ใช้ระบุว่าจะหาค่ากระแสลัดวงจรทั้ง 3 กรณี แต่ถ้าผู้ใช้ต้องการหาค่ากระแสลัดวงจรเฉพาะแบบสามเฟสสมดุลเท่านั้น วินโดว์แสดงข้อมูลจะไม่แสดงค่าอิมพีแดนซ์ลำดับศูนย์ เพราะไม่จำเป็นต้องใช้ในการคำนวณกระแสลัดวงจรประเภทนี้ โดยตัวอย่างของวินโดว์แสดงค่าอิมพีแดนซ์กรณีนี้จะเป็นดังรูปที่ 7.31

สำหรับกระบวนการยุบค่าอิมพีแดนซ์ในวงจรด้วย Complex Calculator (รูปที่ 7.32) โปรแกรมได้ถูกออกแบบให้ผู้ใช้สามารถคำนวณได้อย่างสะดวก โดยใช้ตัวแปรหลัก Z1 และ Z0 ซึ่งมีตัวปฏิบัติการ (Operator) 2 ตัว คือ ปุ่มต่ออนุกรม (Series) และปุ่มต่อขนาน (Parallel)

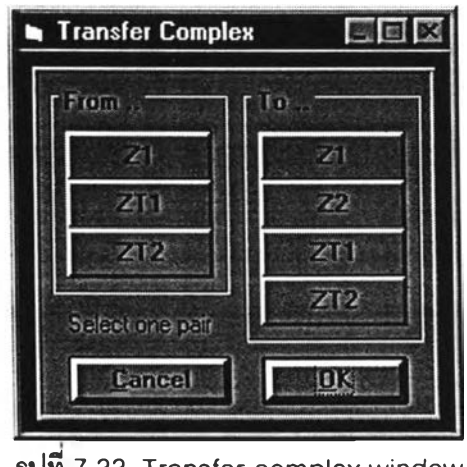


รูปที่ 7.31 Transformer-impedance window ที่แสดงเฉพาะ Z1

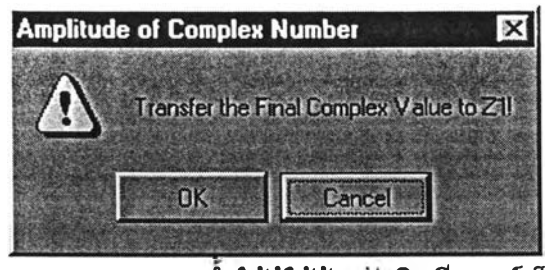


รูปที่ 7.32 Complex Calculator

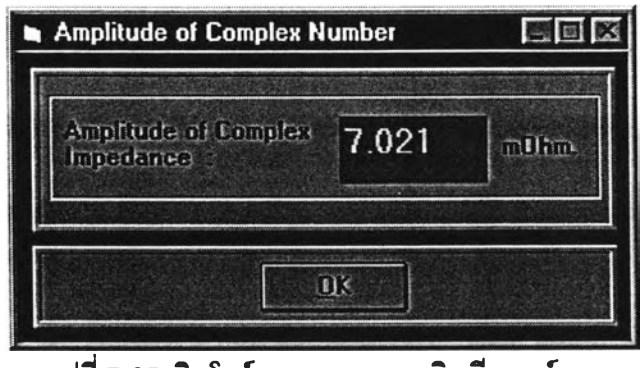
วิธีการใช้ Complex Calculator นี้ เริ่มจากการที่ผู้ใช้เติมค่าอิมพีแดนซ์ Z1 (=R1+jX1) และ Z2 (=R2+jX2) ซึ่งดูค่าอิมพีแดนซ์ของอุปกรณ์แต่ละตัวจากวินโดว์แสดงค่าอิมพีแดนซ์จากการดับเบิลคลิ๊กที่รูปอิมพีแดนซ์บนวงจรตามที่กล่าวไปแล้ว จากนั้นก็กดปุ่มปฏิบัติการ Series หรือ Parallel ตามลักษณะการต่อของอิมพีแดนซ์ทั้งสองตัวว่าจะเป็นแบบต่ออนุกรม หรือต่อแบบขนาน ซึ่งค่าผลรวมของอิมพีแดนซ์ที่ได้หลังจากกดปุ่มปฏิบัติการจะไปแทนที่ค่าในตัวแปร Z1 และด้วยหลักการนี้ ผู้ใช้สามารถยุบวงจรได้ทั้งหมด แต่อย่างไรก็ตาม ในบางกรณีวงจรอาจมีความซับซ้อนมากซึ่งผู้ใช้ก็สามารถย้าย (Transfer) ผลคำนวณอิมพีแดนซ์ที่ได้ในสายวงจรย่อยหนึ่งๆไว้ในตัวแปรชั่วคราว ZT1 (=RT1+jXT1) หรือ ZT2 (=RT2+jXT2) เพื่อคำนวณค่าอิมพีแดนซ์รวมในสายย่อยอื่นๆก่อน ซึ่งการย้ายค่าอิมพีแดนซ์นี้จะกระทำได้ใน Transfer-complex window ดังรูปที่ 7.33 และเมื่อคำนวณค่าอิมพีแดนซ์ในสายย่อยอื่นเสร็จ จึงย้ายค่ากลับไปยังตัวแปรหลักเพื่อใช้ตัวปฏิบัติการในการยุบวงจรอีกครั้งหนึ่ง ซึ่งเมื่อยุบวงจรเรียบร้อยแล้ว และเหลือค่าที่ได้จากการยุบวงจรในตัวแปร Z1 (=R1+jX1) ผู้ใช้ก็จะต้องหาขนาดของอิมพีแดนซ์รวมนี้เพื่อนำไปคำนวณหากระแสลัดวงจรต่อไป โดยโปรแกรมนี้ได้จัดปุ่มพิเศษในการหาขนาดของอิมพีแดนซ์รวมด้วยปุ่ม Amplitude แต่ก่อนที่ส่วนประมวณผลของโปรแกรมจะคำนวณหาขนาดของอิมพีแดนซ์รวม โปรแกรมจะแสดง Message box บอกย้ำผู้ใช้ให้ย้ายค่าอิมพีแดนซ์รวมไปยังตัวแปร Z1 ก่อน ดังรูปที่ 7.34 หลังจากนั้นจึงเข้าสูตรคำนวณหาขนาดของอิมพีแดนซ์รวมได้ โดยผลลัพท์จะถูกแสดงในวินโดว์ Amplitude-of-complex window ดังรูปที่ 7.35



รูปที่ 7.33 Transfer-complex window

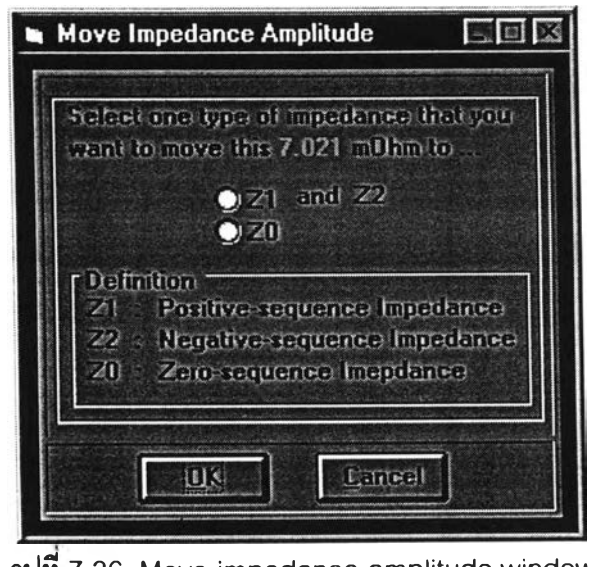


รูปที่ 7.34 Message box บอกย้ำให้ผู้ใช้ย้ายค่าอิมพีแดนซ์เก็บที่ตัวแปร Z1



รูปที่ 7.35 วินโดว์แสดงขนาดของอิมพีแดนซ์รวม

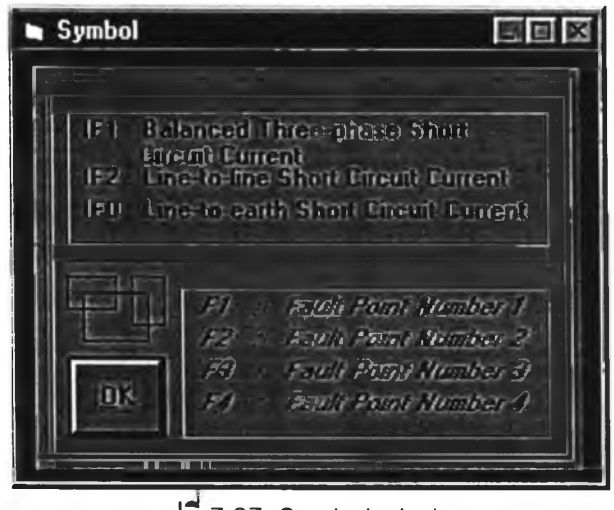
หลังจากที่ผู้ใช้กดปุ่ม OK ในวินโดว์แสดงขนาดของอิมพีแดนซ์รวม (รูปที่ 7.35) ก็จะปรากฏ Move-impedance-amplitude window ดังแสดงด้วยรูปที่ 7.36 ซึ่งช่วยอำนวยความสะดวกให้แก่ผู้ใช้ในการถ่ายโอนค่าผลลัพท์ที่ได้นี้ไปยังช่องรับค่าผลรวมของอิมพีแดนซ์ ซึ่งจะถูกนำไปใช้ในการคำนวณขนาดของกระแสลัดวงจรต่อไป โดยผู้ใช้ต้องระบุด้วยว่าผลลัพท์ที่ได้นี้จะเป็นของ Z1 หรือ Z0



รูปที่ 7.36 Move-impedance-amplitude window

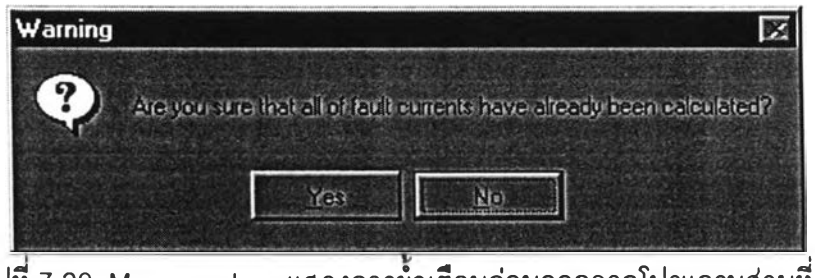
และเมื่อข้อมูลของการคำนวณกระแสลัดวงจรครบทุกตัวแปรแล้ว (Z1, Z2, Z0 สำหรับหากระแสลัดวงจรทั้ง 3 แบบ และ Z1 สำหรับกระแสลัดวงจรเฉพาะแบบสามเฟสสมดุล) ผู้ใช้จึงสามารถสั่งให้หน่วยประมวณผลของโปรแกรมเริ่มการคำนวณกระแสลัดวงจรได้ โดยกดปุ่ม IF1, IF2 และ IF0

เนื่องจากตัวแปรที่ใช้ในการคำนวณกระแสลัดวงจรนี้มีหลายตัว และค่อนข้างซับซ้อน ผู้ออกแบบจึงได้จัดให้มีปุ่มอธิบายความหมายของตัวแปรเหล่านี้ด้วย โดยการกดปุ่ม Symbol บนหน้าจอของโปรแกรมส่วนที่สองนี้ ซึ่งจะมีการแสดง Symbol window ดังรูปที่ 7.37



รูปที่ 7.37 Symbol window

ก่อนที่จะออกจากโปรแกรมในส่วนที่สองนี้ ผู้ใช้ต้องคำนวณกระแสลัดวงจรให้เสร็จในทุกกรณีและทุกจุดผิดพร่องเสียก่อน ซึ่งเมื่อผู้ใช้กดปุ่ม Back เพื่อออกจากโปรแกรมส่วนที่สอง โปรแกรมจะแสดง Message box ดังรูปที่ 7.38 ด้วยข้อความย้ำเตือนก่อนออกจากโปรแกรม



รูปที่ 7.38 Message box แสดงการย้ำเตือนก่อนออกจากโปรแกรมส่วนที่ 2

7.4.3 ส่วนวิเคราะห์และออกแบบระบบไฟฟ้าโดยใช้กราฟลักษณะการทำงานของอุปกรณ์ป้องกัน

วัตถุประสงค์ของโปรแกรมในส่วนนี้ ต้องการให้ผู้ใช้สามารถวิเคราะห์ระบบป้องกันแบบประสานการทำงานได้แทนวิธีเดิมแบบ Manual ที่มีความยุ่งยากในทางปฏิบัติ นอกจากนั้นก็ยังพัฒนาโปรแกรมให้สามารถจัดเก็บงานเก่า และพิมพ์ข้อมูลออกมาใช้เป็นเอกสารในการแสดงผลการวิเคราะห์ได้ด้วย โดยจะมีรายละเอียดของโปรแกรม ดังต่อไปนี้

7.4.3.1 **ลักษณะของโปรแกรมส่วนที่ 3**

- สามารถสร้างกราฟของอุปกรณ์ป้องกัน ซึ่งได้แก่
  1) Electromagnetic circuit breaker
  2) Electronic circuit breaker
  3) Fuse
- สามารถสร้างกราฟของมอเตอร์
- มีตารางแสดงชื่อไฟล์ของสายวงจรที่ถูกวิเคราะห์ (*.PRJ) โดยอ้างอิงกับไฟล์ของไดอะแกรมในโปรแกรมส่วนที่หนึ่งที่มีนามสกุล OCP พร้อมทั้งมีการบอกชื่ออุปกรณ์ป้องกันที่เกี่ยวข้องตามลำดับการตัดวงจร ณ.จุดผิดพร่องของไฟล์ที่กำลังพิจารณาด้วย
- แสดง Inrush point และ Transformer point ของหม้อแปลง
- กำหนดชื่อของอุปกรณ์ป้องกันทั้งหมด
- กำหนดชื่อของมอเตอร์ และหม้อแปลง
- กำหนดชื่อของจุดที่เกิดกระแสผิดพร่อง
- รูปกราฟของเซอร์กิตเบรคเกอร์สามารถเลื่อน และปรับเปลี่ยนได้
- เปลี่ยนขนาดพิกัดของมอเตอร์ได้โดยไม่ต้องสร้างกราฟใหม่
- ลบกราฟที่ไม่ต้องการทิ้งได้
- โหลดกราฟของอุปกรณ์ป้องกันทุกชนิด และมอเตอร์ได้โดยตรงจากฐานข้อมูล (Database)
- มีข้อความย้ำเตือนผู้ใช้ถึงข้อผิดพลาดที่อาจเกิดขึ้นได้ในระหว่างการวิเคราะห์การวางตัวของเส้นโค้งอุปกรณ์
- เปลี่ยนขนาดความหนาของกราฟได้
- เปลี่ยนชนิด และสีของเส้นกราฟต่างๆได้ตามต้องการ
- เปลี่ยนสีของหน้าจอ และตัวอักษรได้

- จัดเก็บ (Save) กราฟของอุปกรณ์ป้องกัน และมอเตอร์ แล้วสามารถเรียกมาดัดแปลงแก้ไขในภายหลังได้

- พิมพ์กราฟของอุปกรณ์ป้องกันต่างๆ, กราฟของมอเตอร์, เส้นกำหนดกระแสผิดพร่อง, Inrush point และ Transformer point ของหม้อแปลงทางเครื่องพิมพ์ได้

- มีคำอธิบายการใช้งานของโปรแกรม (Help)

- มีการเชื่อมต่อกับโปรแกรมในส่วนที่ 1 (โดยกดปุ่ม Back บน Main window ซึ่งจะนำไปสู่ Single-line-diagram window ของโปรแกรมในส่วนแรก)

- เช่นเดียวกับโปรแกรมส่วนที่ 1 และ 2 รูปภาพบนหน้าจอที่ไม่ได้แสดงผลจะถูกลบออก (Unload) เพื่อความประหยัดต่อหน่วยความจำของคอมพิวเตอร์

7.4.3.2 รูปแบบของโปรแกรมส่วนที่ 3

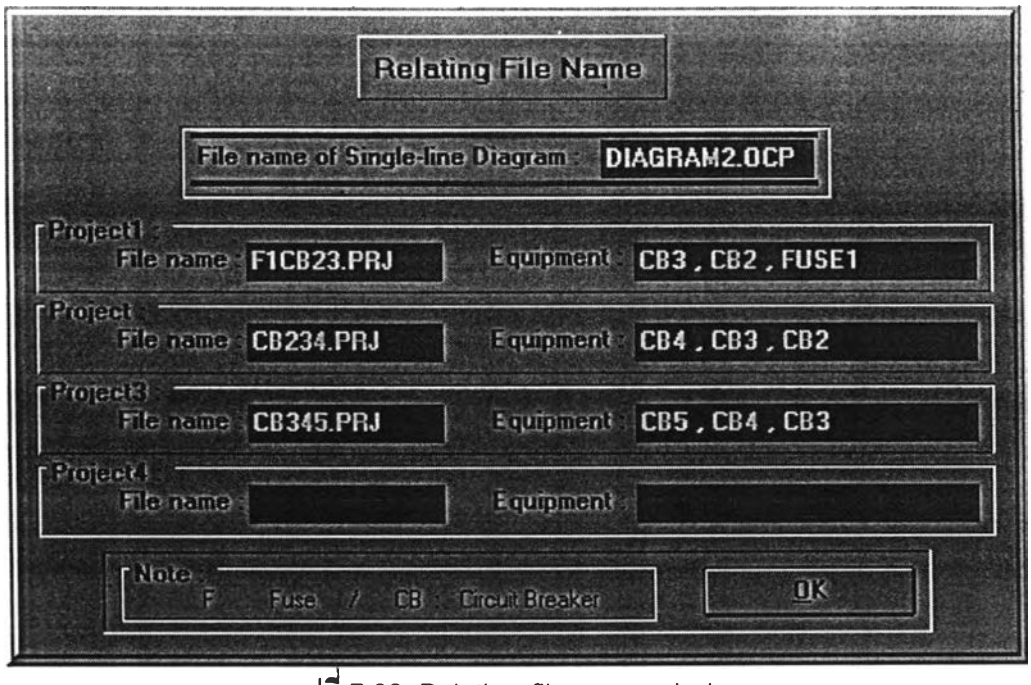
หลังจากเขียนไดอะแกรมเส้นเดี่ยว และคำนวณกระแสลัดวงจรเสร็จแล้ว เมื่อกดปุ่ม Next ใน Single-line-diagram window ก็จะเป็นการเข้าสู่โปรแกรมในส่วนที่ 3 ซึ่งจะเริ่มเมื่อผู้ใช้เรียกงานเก่าด้วยคำสั่ง Open ใน File menu จากนั้นจะปรากฏวินโดว์แสดงชื่อไฟล์และชื่ออุปกรณ์ป้องกันที่เกี่ยวข้องกับไดอะแกรมที่สร้างขึ้นในส่วนแรก (Relating-file-name window) เพื่อเป็นตัวอ้างอิงให้ผู้ใช้รู้ว่า สำหรับไฟล์ของไดอะแกรมในส่วนแรกนั้น มีกี่ไฟล์ที่ใช้ในการวิเคราะห์ลักษณะเส้นโค้งของอุปกรณ์ป้องกัน และไฟล์เหล่านั้นมีชื่ออะไรบ้าง เพื่อให้ผู้ใช้ย้อนไปดูงานเก่าได้ เพราะไดอะแกรมหนึ่งๆสามารถมีได้หลายสายวงจรที่ถูกวิเคราะห์ดังตัวอย่างรูปที่ 7.39 ซึ่งหมายความว่ามีไฟล์ในการวิเคราะห์เส้นโค้ง 3 ไฟล์ ที่ชื่อ F1CB23.PRJ, CB234.PRJ และ CB345.PRJ สำหรับไฟล์ของไดอะแกรม DIAGRAM2.OCP ในโปรแกรมส่วนที่ 1 โดยไฟล์แรกเป็นการวิเคราะห์ลักษณะเส้นโค้งของ FUSE1, CB2 และ CB3 ส่วนไฟล์ที่สองเป็นการวิเคราะห์ของอุปกรณ์ CB2, CB3 และ CB4 และเมื่อผู้ใช้กดปุ่ม OK ในวินโดว์นี้ก็จะปรากฏ วินโดว์หลัก (Main window) ดังที่แสดงด้วยรูปที่ 7.40 ซึ่งภายใน Main window นี้จะประกอบด้วยกราฟสเกล Log-Log โดยมีแกน X เป็นกระแสที่มีหน่วยเป็นแอมแปร์ (Ampere) ตั้งแต่ 10 ถึง 1000 กิโลแอมแปร์ ส่วนแกน Y เป็นเวลา ที่มีหน่วยเป็นวินาที ตั้งแต่ 0.01 ถึง 1000 วินาที

แต่ในกรณีที่ผู้ใช้เริ่มงานใหม่ด้วยคำสั่ง New ใน File menu ลำดับการทำงานของโปรแกรมในส่วนที่ 3 จะต่างไปจากที่กล่าวมาแล้วข้างต้น นั่นคือจะเริ่มจากการที่ผู้ใช้เรียกเส้นโค้งของอุปกรณ์ป้องกันที่เกี่ยวข้องพร้อมรายละเอียดเสริมอื่นๆ ซึ่งได้แก่

- จุด Inrush point และ ANSI point เมื่อมีการใช้ฟิวส์ป้องกันวงจรด้านแรงดันระดับสูงของหม้อแปลง

- เส้นกำหนดกระแสลัดวงจร

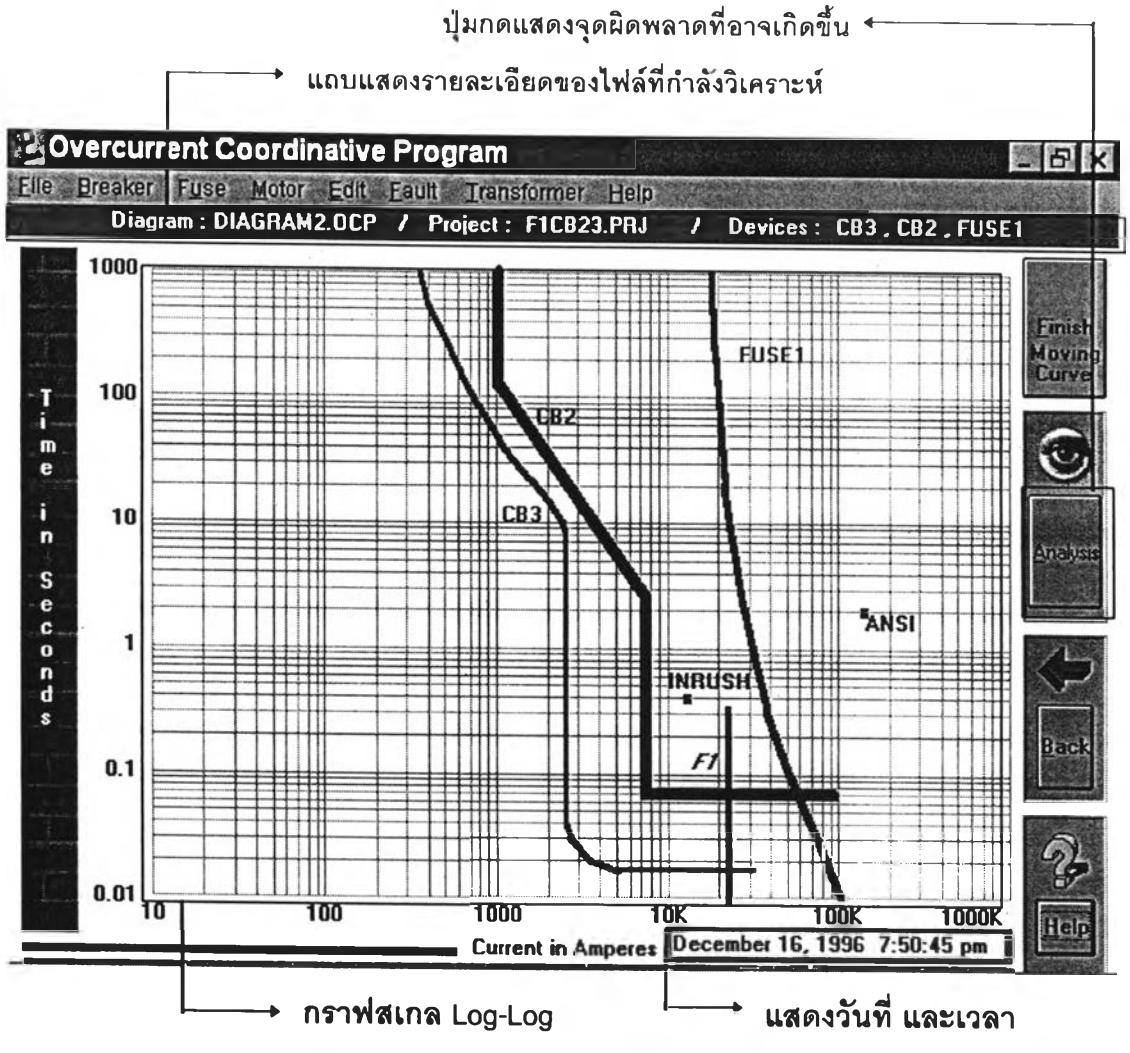
จากนั้นผู้ใช้ก็ต้องกดปุ่ม Analysis เพื่อทำงานในลำดับถัดไป



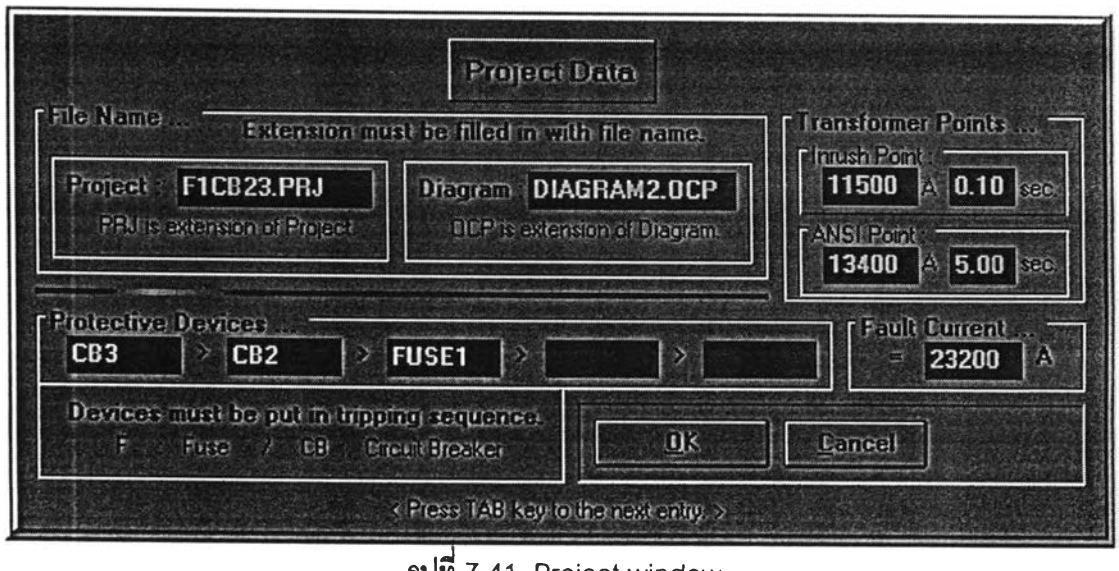
รูปที่ 7.39 Relating-file-name window

หลังจากที่ผู้ใช้กดปุ่ม Analysis โปรแกรมจะแสดง Save control box เพื่อให้ผู้ใช้จัดเก็บชื่องานที่จะวิเคราะห์ก่อน จากนั้นผู้ใช้จะต้องกรอกข้อมูลของไฟล์ที่จะวิเคราะห์ใน Project window (รูปที่ 7.41) ซึ่งข้อมูลที่กล่าวมานี้ คือชื่อของไฟล์, ขนาดของกระแสผิดพร่อง และอุปกรณ์ตัดวงจรที่เกี่ยวข้องในสายวงจรที่กำลังจะวิเคราะห์ โดยอุปกรณ์ป้องกันนั้นต้องเรียงลำดับการตัดวงจรด้วย ซึ่งอุปกรณ์ป้องกันที่จะตัดวงจรเร็วที่สุดในที่นี้ หมายถึงอุปกรณ์ที่อยู่ในสายล่างที่สุดของส่วนวงจรที่กำลังวิเคราะห์ คือจะเป็นอุปกรณ์ที่ทำงานเป็นลำดับแรกของสายวงจรนั้นๆ

พิจารณาจาก Project window รูปที่ 7.41 ซึ่งเป็นตัวอย่างการลงรายละเอียดข้อมูลของไฟล์ F1CB23.PRJ โดยจะเห็นว่าลำดับการตัดวงจรของอุปกรณ์ป้องกันในที่นี้ คือ CB3, CB2 และ FUSE1 ตามลำดับ ซึ่งข้อมูลที่สำคัญใน Project window จะถูกแสดงออกมาที่แถบแสดงรายละเอียดของไฟล์ที่จะวิเคราะห์บน Main window เพื่อใช้ในการวิเคราะห์งาน (ดังในรูปที่ 7.40)



รูปที่ 7.40 Main window ของโปรแกรมในส่วนที่ 3



รูปที่ 7.41 Project window

หลังจากที่ผู้ใช้ลงรายละเอียดของไฟล์นั้นๆ เสร็จแล้ว โปรแกรมจะแสดง Error message box บอกย้ำจุดที่มักเกิดข้อผิดพลาดในการวิเคราะห์ ซึ่งจะกล่าวต่อไปในส่วนของการวิเคราะห์เส้นโค้งของอุปกรณ์ป้องกัน

การทำงานของโปรแกรมในส่วนที่ 3 สามารถแบ่งได้เป็น 3 ส่วนย่อย คือ

- **ส่วนของแถบเมนู** (Pop-up menu)
- **ส่วนจัดการกับกราฟของอุปกรณ์** (Curve manipulation)
- **ส่วนย้ำเตือนการวิเคราะห์เส้นโค้งของอุปกรณ์ป้องกัน** (The emphasis of curve analysis)

โดยในแต่ละส่วนจะมีรายละเอียด ดังต่อไปนี้

3.1) ส่วนของแถบเมนู

ภายใน Main window **จะมีแถบเมนู** (Menu bar) **ดังที่เห็นในรูปที่** 7.40 **ซึ่งผู้ใช้โปรแกรมสามารถเลือกใช้ตัวเลือกของคำสั่งต่างๆ ซึ่งประกอบไปด้วย**

3.1.1) File menu (**หรือกด** ALT-F)

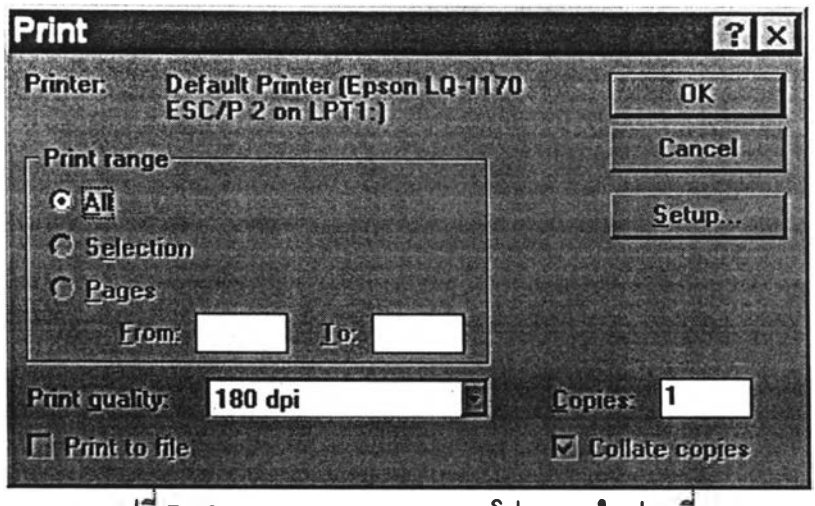
- New option : **สร้างโครงงาน** (Project) **ใหม่ โดยกราฟที่มีอยู่เดิมจะถูกลบทิ้ง**
- Open option : **เหมือนกับคำสั่ง** New **แต่จะเป็นการเปิดไฟล์ที่มีอยู่แล้วขึ้นมาแก้ไขดัดแปลง**
- Save option : **สำหรับจัดเก็บ** Project **ที่กำลังใช้งานลงไปในไฟล์**
- Print option : **พิมพ์งานออกมาทางเครื่องพิมพ์ ซึ่งมีกรอบควบคุมดังรูปที่** 7.42 **โดยสามารถทำได้** 2 **แบบ คือ**
  - **แบบครึ่งหน้า** (Half page) : **พิมพ์กราฟในขนาดครึ่งหน้าของกระดาษขนาด** A4
  - **แบบเต็มหน้า** (Full page) : **พิมพ์กราฟในขนาดเต็มหน้าของกระดาษขนาด** A4
- Exit option : **เพื่อออกจากโปรแกรม**

3.1.2) Breaker menu (**หรือกด** ALT-B)

- New option : **เปิด** New MCB **หรือ** New ECB window
- Open option : **เปิด** Open MCB **หรือ** Open ECB window

3.1.3) Fuse menu (**หรือกด** ALT-F)

- New option : **เปิด** New fuse-curve window
- Open option : **เปิด** Open fuse window



รูปที่ 7.42 Print control box ของโปรแกรมในส่วนที่ 3

3.1.4) Motor menu (หรือกด ALT-M)

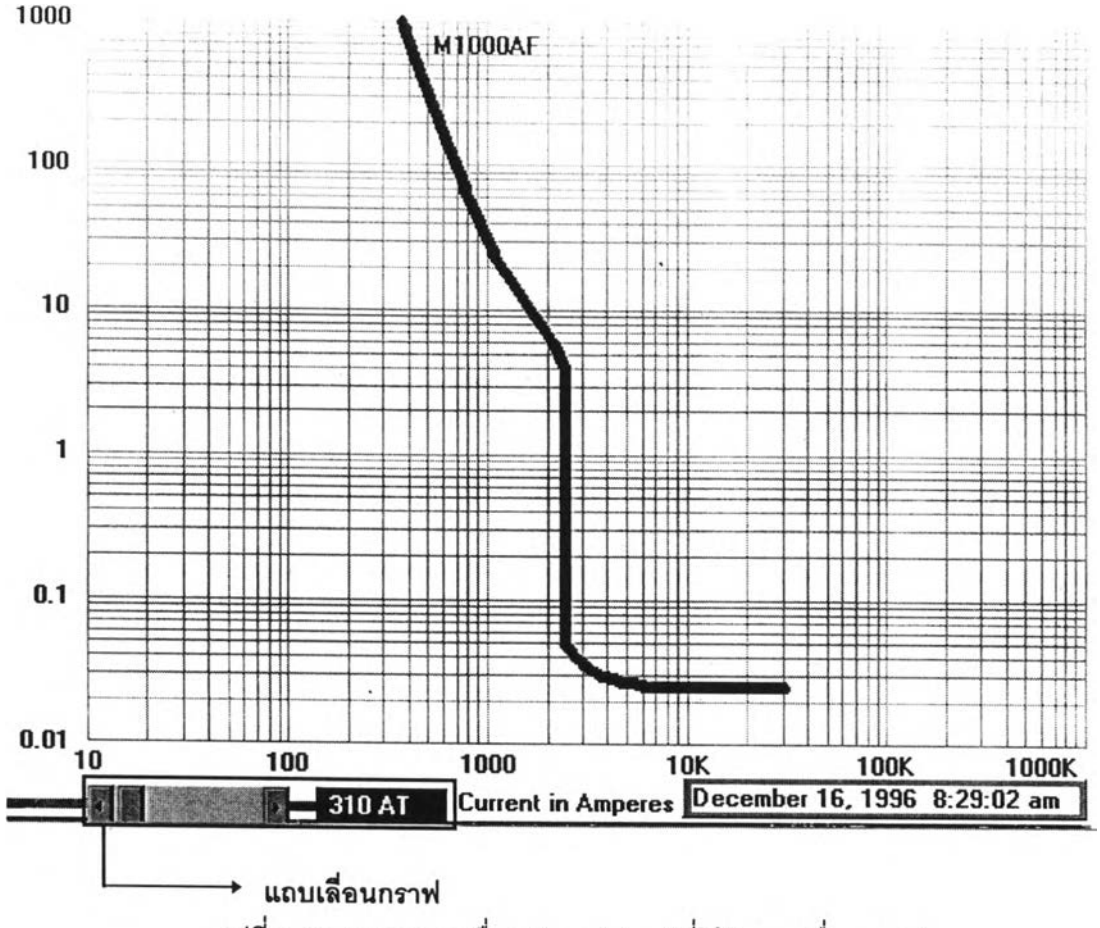
- New option : เปิด New motor-curve window
- Open option : เปิด Open motor window
- Size option : เปลี่ยนขนาดกระแสพิกัดของมอเตอร์

3.1.5) Edit menu (หรือกด ALT-E)

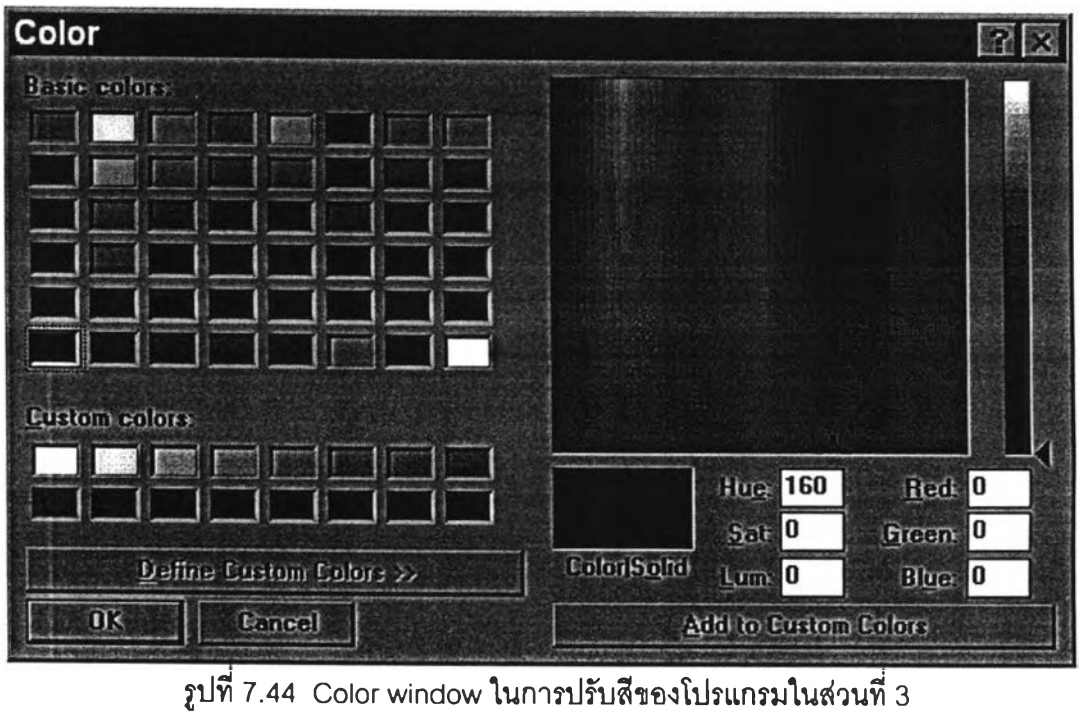
- Move option : เลื่อนกราฟของเซอร์กิตเบรคเกอร์ในช่วงตั้งแต่ Ampere trip ที่ต่ำที่สุด ถึงขนาดของ Ampere frame (โดยตั้งค่าเหล่านี้ได้ใน New MCB และ New ECB window) ซึ่งหลังจากกดที่แถบคำสั่งนี้ โปรแกรมจะจัดแถบเลื่อน (Scroll bar) ให้ผู้ใช้สามารถเลื่อนกราฟนั้นๆได้ในแนวระดับดังที่แสดงด้วยรูป 7.43 และเมื่อทำการเลื่อนกราฟเสร็จแล้ว ค่า Ampere trip จะถูกปรับเปลี่ยนโดยอัตโนมัติ (เมื่อผู้ใช้เลื่อนกราฟเสร็จแล้วจะต้องกดปุ่ม Finish Moving Curve ที่อยู่บน Main window ด้วย)

นอกจากนั้น ชื่อของอุปกรณ์ป้องกันจะสามารถเปลี่ยนตำแหน่งไป ณ.ตำแหน่งใดก็ได้บนกราฟสเกล Log-Log โดยการใช้เมาส์คลิ๊กที่ชื่อนั้น ซึ่งจะมีกรอบสีเทาให้ผู้ใช้ลากไปในตำแหน่งที่ต้องการ

- Delete option : ใช้ลบกราฟที่ไม่ต้องการ
- Color option : เปลี่ยนสีของกราฟอุปกรณ์ (ซึ่งได้แก่ เซอร์กิตเบรคเกอร์, ฟิวส์ และมอเตอร์), เส้นกราฟสเกล Log-Log, หน้าจอ และตัวอักษรที่บอกแกน โดยผู้ใช้สามารถเลือกสีได้มากมาย ดังที่เห็นใน Color window ของรูปที่ 7.44



รูปที่ 7.43 แสดงแถบเลื่อน (Scroll bar) ที่ใช้ในการเลื่อนกราฟ



รูปที่ 7.44 Color window ในการปรับสีของโปรแกรมในส่วนที่ 3

- Width option : เปลี่ยนขนาดเส้นกราฟของอุปกรณ์

- Style option : เปลี่ยนลักษณะเส้นกราฟของอุปกรณ์ ซึ่งทำได้ 5 แบบ คือ เส้นทึบ และ เส้นประ 4 แบบ

3.1.6) Fault menu (หรือกด ALT-F)

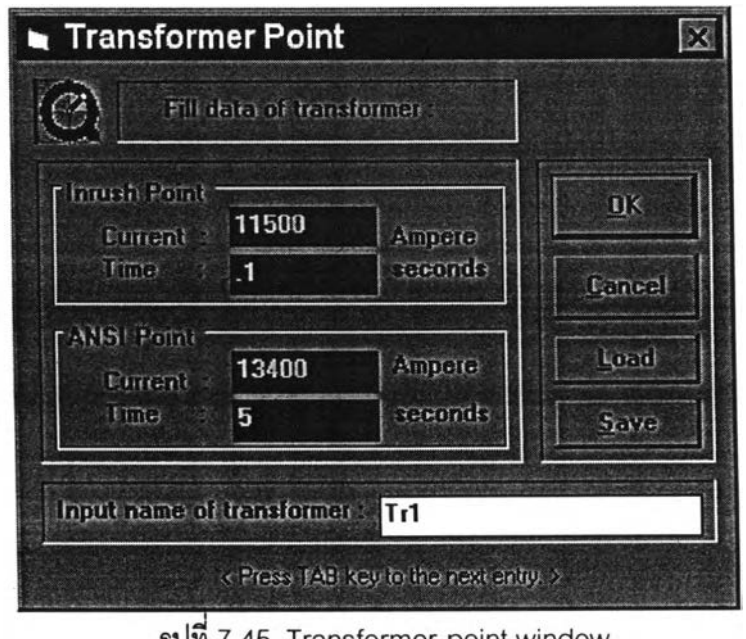
- Show option : แสดงเส้นกำหนดขนาดของกระแสผิดพร่อง ซึ่งได้ผลมาจากการคำนวณของโปรแกรมในส่วนที่ 2 หรืออาจกำหนดค่าเองก็ได้

- Clear option : ลบกระแสผิดพร่องทิ้งไปเพียงบางตัว หรือลบทิ้งทั้งหมดก็ได้

3.1.7) Transformer menu (หรือกด ALT-T)

- Show option : แสดง Transformer-point window ดังรูปที่ 7.45 ซึ่งเป็นวินโดว์ที่ใช้ในการกำหนด Inrush point และ ANSI point ของหม้อแปลง รวมทั้งการกำหนดชื่อและรายละเอียดของหม้อแปลง ซึ่งผู้ใช้สามารถดึงข้อมูลเก่าขึ้นมาแก้ไข และเก็บข้อมูลปัจจุบันไว้ในไฟล์ได้ ด้วยคำสั่ง Load และ Save ตามลำดับ

- Clear option : ลบ Inrush point และ ANSI point ทุกจุด



รูปที่ 7.45 Transformer-point window

3.1.8) Help menu (หรือกด ALT-H)

โปรแกรมในส่วนที่สามนี้ ได้จัดให้มีคำอธิบายการใช้โปรแกรมดังแสดงไว้ในภาคผนวก ก.

3.2) ส่วนจัดการกับกราฟของอุปกรณ์

โปรแกรมในส่วนที่ 3 ได้จัดให้มีส่วนที่จัดการกับกราฟของอุปกรณ์ คือการสร้างกราฟใหม่ และโหลดกราฟที่สร้างไว้แล้วมาใช้ หรือดัดแปลงแก้ไข ซึ่งอุปกรณ์ป้องกันที่สามารถทำได้มีอยู่ 3 ประเภท คือ

- เซอร์กิตเบรคเกอร์ (Circuit breaker)
- ฟิวส์ (Fuse)
- มอเตอร์ (Motor)

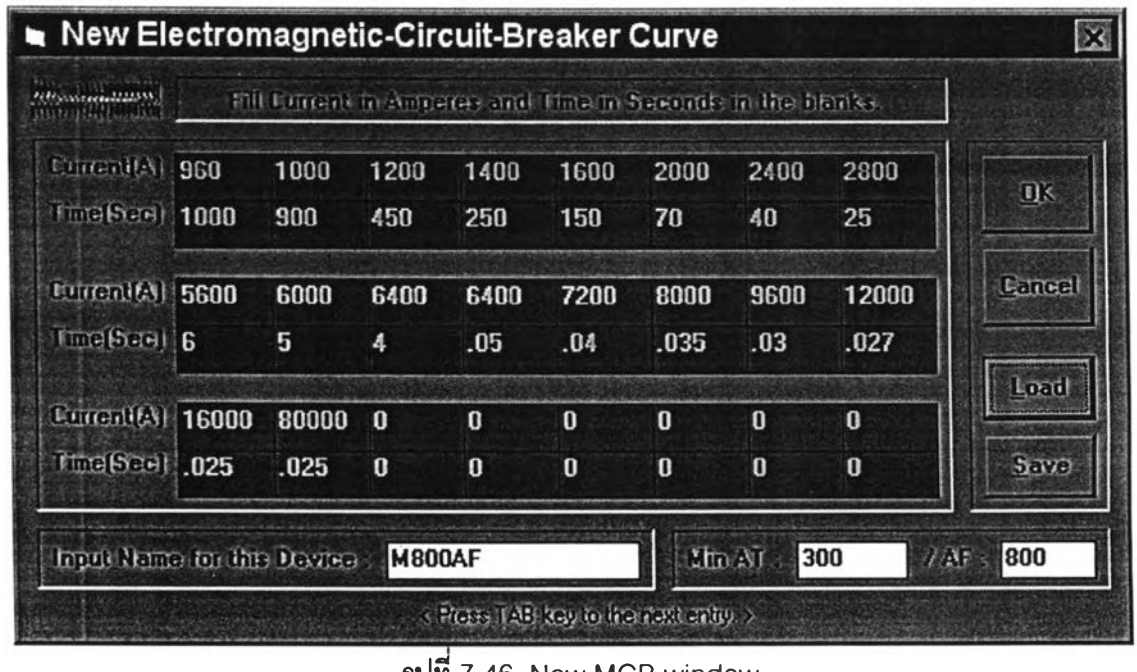
โดยอุปกรณ์แต่ละประเภท จะมีรายละเอียด ดังนี้

1) **เซอร์กิตเบรคเกอร์** : สามารถแบ่งได้เป็น 2 ชนิด คือ

1.1) Electromagnetic circuit breaker

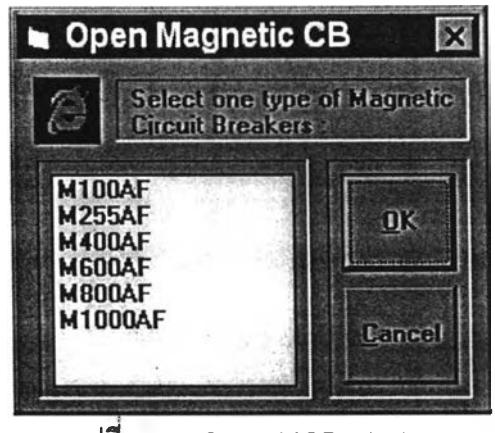
ผู้ใช้โปรแกรมสามารถสร้างกราฟของเซอร์กิตเบรคเกอร์แบบนี้โดยการกำหนดจุดได้ถึง 24 จุด ในช่องที่กำหนดไว้ให้ของรูปที่ 7.46 จากนั้นให้ใส่ชื่อของเซอร์กิตเบรคเกอร์ และขนาด Ampere trip ที่น้อยที่สุดที่อุปกรณ์นั้นจะถูกปรับตั้งได้ รวมถึงขนาดของ Ampere frame ซึ่งจะเป็นตัวกำหนดขอบเขตในการเลื่อนกราฟ

นอกจากนั้น ถ้าต้องการเรียกข้อมูลเก่าที่ทำไว้แล้วขึ้นมาแก้ไข ก็สามารถทำได้โดยใช้คำสั่ง Load และถ้าต้องการเก็บบันทึกข้อมูลลงไปเป็นไฟล์ ก็สามารถทำได้โดยใช้คำสั่ง Save ที่อยู่ในวินโดว์นี้



รูปที่ 7.46 New MCB window

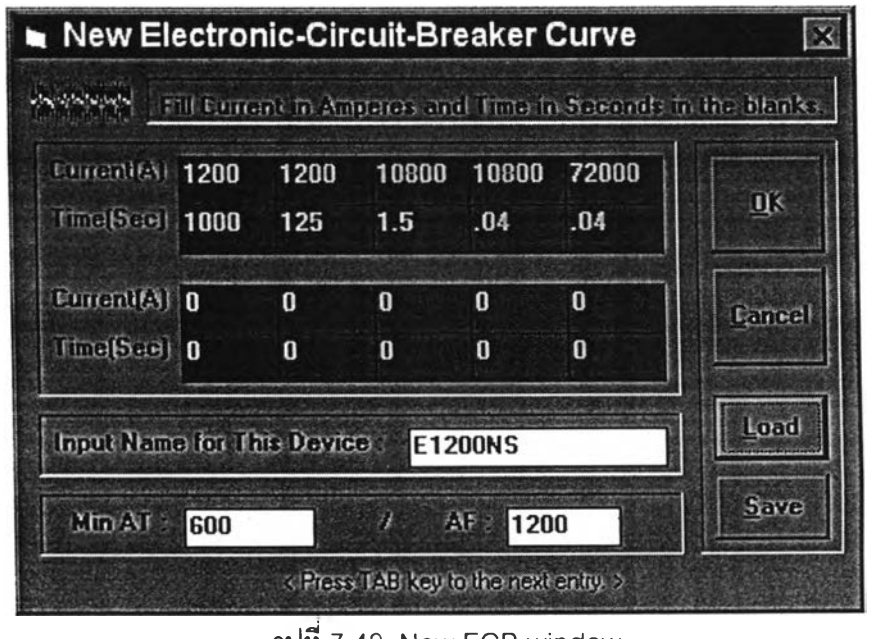
ในกรณีที่ผู้ใช้ต้องการโหลดกราฟของเซอร์กิตเบรคเกอร์แบบนี้จากฐานข้อมูล (Database) ก็สามารถทำได้โดยดับเบิลคลิ๊กที่ชื่อของเซอร์กิตเบรคเกอร์ที่ต้องการ แทนการกดปุ่ม OK ดังที่แสดงในรูปที่ 7.47 โดยที่ผู้ใช้สามารถโหลดได้มากกว่า 1 กราฟ ในงานหนึ่งๆ



รูปที่ 7.47 Open MCB window

1.2) Electronic circuit breaker

เช่นเดียวกับ Electromagnetic circuit breaker ผู้ใช้โปรแกรมสามารถสร้างกราฟของอุปกรณ์ป้องกันประเภทนี้ได้โดยการใส่จุดคู่ลำดับ แต่เนื่องจากกราฟของ Elecctronic circuit breaker มีความละเอียดหรือความโค้งน้อยกว่า Electromagnetic circuit breaker ดังนั้นจึงสร้างกราฟได้ด้วยคู่ลำดับเพียง 10 จุดเท่านั้น ดังแสดงด้วยรูปที่ 7.48



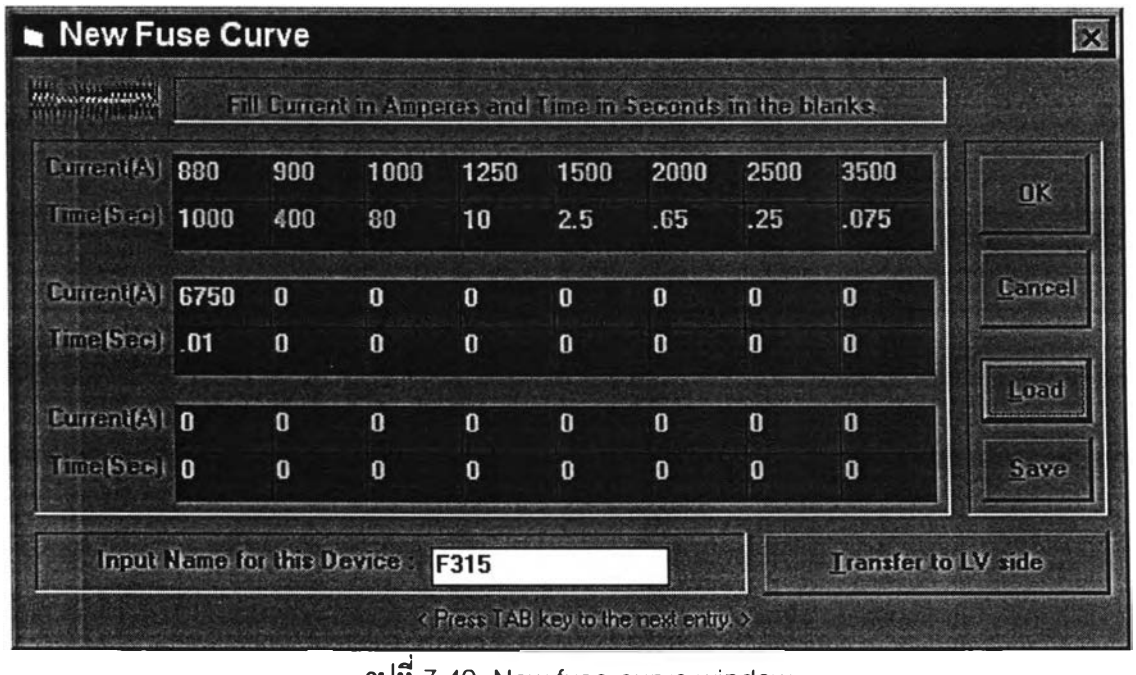
รูปที่ 7.48 New ECB window

ส่วนการโหลดกราฟของ Electronic circuit breaker นั้น จะเป็นเช่นเดียวกับกรณีของ Electromagnetic circuit breaker

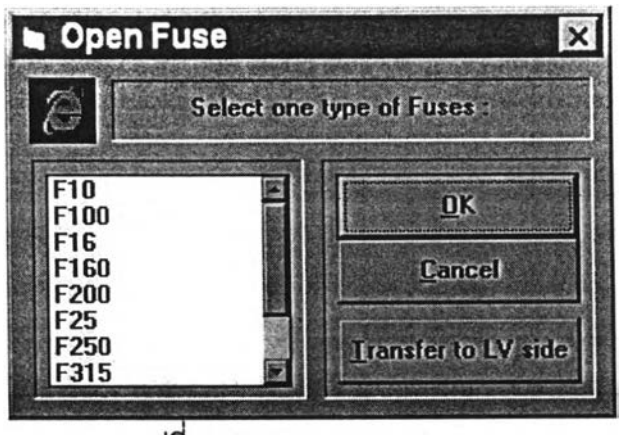
2) **ฟิวส์**

ผู้ใช้สามารถสร้างกราฟของฟิวส์ โดยกำหนดจุดได้ถึง 24 จุด และในกรณีที่ใช้ฟิวส์ป้องกันหม้อแปลงในระบบไฟฟ้า แล้วต้องการย้ายค่ากราฟกระแสจากด้านแรงดันสูงไปยังด้านแรงดันต่ำ ก็สามารถทำได้โดยกดปุ่ม **Transfer to LV side** ดังที่ปรากฏบน New fuse-curve window ของรูปที่ 7.49

ส่วนการโหลดกราฟของฟิวส์ที่สร้างเก็บไว้ในฐานข้อมูลนั้น กราฟที่ได้จะยังไม่มีการย้ายไปทางด้านแรงดันต่ำ แต่ในกรณีที่ผู้ใช้ต้องการย้ายกราฟไปสู่ด้านแรงดันระดับต่ำ ก็สามารถทำได้เช่นกันโดยกดปุ่ม **Transfer to LV side** ใน Open fuse window แทนปุ่ม OK ดังที่ปรากฏบนรูปที่ 7.50



รูปที่ 7.49 New fuse-curve window

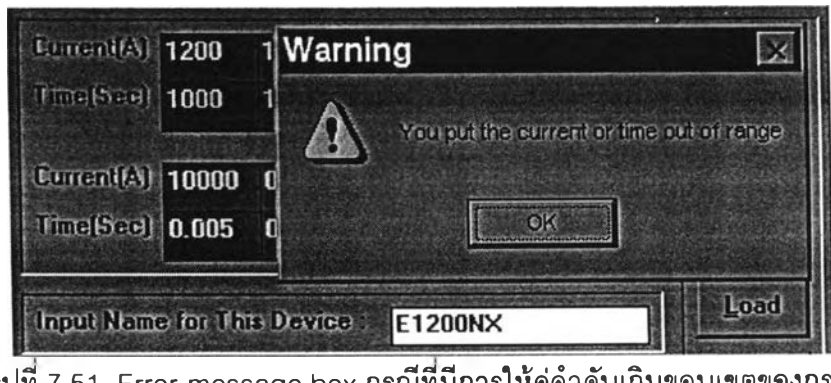


รูปที่ 7.50 Open fuse window

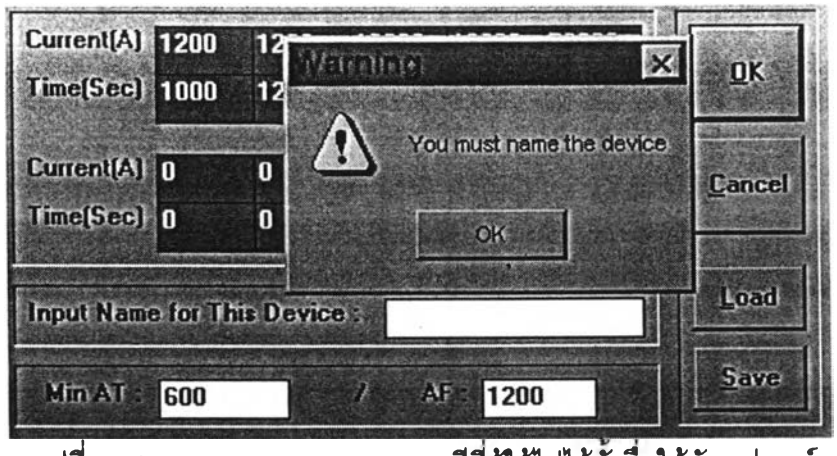
3) **มอเตอร์**

ผู้ใช้โปรแกรมสามารถสร้างกราฟลักษณะการทำงานของมอเตอร์ได้เช่นเดียวกับกรณีของ Electromagnetic circuit breaker คือลงรายละเอียดได้ถึง 24 จุด และก็ยังสามารถจัดเก็บ หรือเปิดกราฟที่ถูกบันทึกไว้ในฐานข้อมูลมาใช้ต่อได้เช่นกัน

ในการสร้างกราฟของอุปกรณ์เหล่านี้ บางกรณีก็อาจมีความผิดพลาดเกิดขึ้นได้ ซึ่งโปรแกรมในส่วนนี้ได้ถูกออกแบบให้มีความสามารถในการแจ้งถึงความผิดพลาดนั้นๆด้วย เพื่อให้ผู้ใช้ใส่ข้อมูลได้อย่างถูกต้อง โดยข้อผิดพลาดที่มักเกิดขึ้น ได้แก่ การให้จุดคู่ลำดับเกินขอบเขตของกราฟ และการที่ผู้ใช้ไม่ได้ตั้งชื่อให้กับอุปกรณ์ ซึ่ง Error-message box ของทั้ง 2 กรณีนี้จะเป็นดังรูปที่ 7.51 และ 7.52 ตามลำดับ



รูปที่ 7.51 Error-message box กรณีที่มีการให้คู่ลำดับเกินขอบเขตของกราฟ



รูปที่ 7.52 Error-message box กรณีที่ผู้ใช้ไม่ได้ตั้งชื่อให้กับอุปกรณ์

3.3) ส่วนย้ำเตือนการวิเคราะห์เส้นโค้งของอุปกรณ์ป้องกัน

หลังจากที่ผู้ใช้เลือกสายวงจรวิเคราะห์ และจัดกลุ่มของเส้นโค้งลักษณะการทำงานของอุปกรณ์ที่เกี่ยวข้องพร้อมทั้งเส้นกำหนดขนาดกระแสลัดวงจร และจุด Inrush point กับจุด ANSI point (ในกรณีที่มีหม้อแปลงในระบบไฟฟ้า) โปรแกรมจะมีการย้ำเตือนถึงข้อผิดพลาดที่มักเกิดขึ้นในการวางตัวของเส้นโค้งอุปกรณ์ป้องกัน ซึ่งการย้ำเตือนนี้จะเกิดจากการที่ผู้ใช้กดปุ่ม Analysis บน Main window

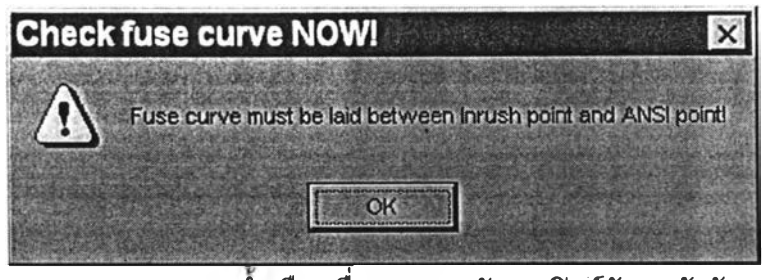
เกณฑ์การวิเคราะห์ในที่นี้จะยึดหลักอยู่ 2 ข้อ คือ

1) กราฟของฟิวส์ต้องอยู่ระหว่าง Inrush point และ ANSI point

ถ้ากราฟของฟิวส์อยู่เหนือจุด ANSI point อาจทำให้ฟิวส์ที่เลือกไม่สามารถป้องกันหม้อแปลงได้ เพราะมีขนาดพิกัดการทำงานที่สูงเกินไป นั่นคือฟิวส์อาจไม่ตัดวงจรเมื่อตรวจจับขนาดกระแสที่ก่อให้เกิดความร้อนเกินกว่าที่หม้อแปลงจะทนได้ ดังนั้นซึ่งผู้ออกแบบต้องเลือกขนาดพิกัดของฟิวส์ให้มีขนาดเล็กลง

แต่ในทางทางตรงกันข้าม ถ้าผู้ใช้เลือกขนาดพิกัดของฟิวส์ต่ำเกินไป ทำให้กราฟของฟิวส์อยู่ใต้จุด Inrush point ซึ่งจะส่งผลให้ฟิวส์ทำการตัดวงจรเร็วเกินไป แม้ว่าจะเป็นการเริ่มเดินเครื่องหม้อแปลงก็ตาม ดังนั้นผู้ออกแบบต้องเลือกขนาดพิกัดของฟิวส์ให้มากขึ้นจากเดิม

จากลักษณะการวางตัวของกราฟฟิวส์เทียบกับจุดทั้ง 2 ตามที่กล่าวมาแล้วนั้น โปรแกรมจะแสดง Message box ดังรูปที่ 7.53 ออกมาเตือนผู้ใช้ให้ระวังในเกณฑ์การวิเคราะห์นี้



รูปที่ 7.53 Error-message box ย้ำเตือนเรื่องการวางตัวของฟิวส์กับจุดบังคับของหม้อแปลง

2) ลำดับการตัดกันระหว่างกราฟของอุปกรณ์ป้องกัน และเส้นตรงกำหนดขนาดของกระแสลัดวงจรที่คำนวณได้ ต้องเป็นไปตามลำดับจากอุปกรณ์ป้องกันตัวล่างของไดอะแกรมที่มีขนาดพิกัดน้อยกว่า ไปยังอุปกรณ์ป้องกันตัวบนของไดอะแกรมที่มีขนาดพิกัดมากกว่า เพื่อให้เกิดสภาพแยกแยะการตัดวงจร (Selectivity) โดยโปรแกรมจะจัดให้มี Message box ออกมาย้ำถึงเรื่องลำดับการตัดวงจรของอุปกรณ์ในสายวงจรที่ถูกวิเคราะห์ ณ.ขณะนั้น ซึ่งจะตรวจลำดับของอุปกรณ์ได้จากแถบข้อมูลของไฟล์ที่ได้แถบเมนูบน Main window และถ้าลำดับการตัดกันของเส้นโค้งไม่เป็นไปตามลำดับที่วางไว้ ผู้ออกแบบจะต้องเปลี่ยน หรือปรับกราฟเดิมให้ถูกต้องก่อนจัดเก็บงาน (Save)

รูปที่ 7.54 Error-message box ย้ำลำดับการตัดวงจรที่ถูกต้องของสายวงจรวิเคราะห์