บทที่ 9

บทสรุปและข้อเสนอแนะ

9.1 สรุปผลงาน

9.1.1 ลักษณะที่ได้

ผลจากการพัฒนาทำให้ได้ซอฟต์แวร์สำหรับการพัฒนาโปรแกรมภาษาฟังก์ชั่นชาร์ต บนไมโครคอมพิวเตอร์ สำหรับเครื่องควบคุมชนิดโปรแกรมได้ (PC) และโดยเฉพาะอย่างยิ่งช่วยให้ PC ที่ไม่รู้จักภาษาฟังก์ชั่นชาร์ต สามารถพัฒนาโปรแกรมเป็นภาษาฟังก์ชั่นชาร์ตได้ ซอฟต์แวร์ที่พัฒนาได้มีคุณสมบัติดังนี้

1. ใช้เขียนโปรแกรมภาษาฟังก์ชั่นชาร์ตได้ โดยเป็นกราฟฟิกเอดิเตอร์ (Graphic Editor) กล่าวคือ สามารถป้อนโปรแกรมเป็นรูปภาพได้โดยตรง

2. สามารถแปลโปรแกรมภาษาฟังก์ชั่นชาร์ตที่เขียน ไปเป็นสมการบูลลีน (Boolean Equation) , ภาษานีมอนิค (Mnemonic Language) ของ PC และภาษาเครื่อง (Internal Code) ของออมร่อน (C-series) ได้

3. สามารถส่งโปรแกรมที่แปลแล้ว (ภาษาเครื่อง) จากไมโครคอมพิวเตอร์ไปยัง PC ของออมร่อน (C-series) ได้ โดยผ่านทางพอร์ตอนุกรม และใช้ RS-232C

4. สามารถทำเอกสารได้ดังต่อไปนี้

   - FUNCTION CHART : ภาพชาร์ต
   - EQUATION OF RELATION : สมการบูลลีนที่แปลได้
   - STEP & TRANSITION LIST : ตารางแสดงการใช้สเต็ปและทรานสิชั่น
   - GENERATED MNEMONIC : โปรแกรมนีมอนิคที่แปลได้
   - I/O & AUXILIARY RELAY MAPPING : ตารางแสดงการนำรีเลย์ช่วยมาใช้งาน

5. สามารถปรับปรุงเพิ่มเติมให้แปลเป็นภาษานีมอนิคของ PC อื่นได้อีกโดยง่ายและไม่จำกัด โดยการเพิ่มเติมชุดคำสั่งของ PC ที่ต้องการ เข้าไปในแฟ้ม "BRANDS.MNE" ตามกฎเกณฑ์ที่กำหนดไว้

6. ในการเขียนชาร์ต การเลือกเส้นทางการทำงาน ณ จุดใดจุดหนึ่ง สามารถเลือกได้สูงสุด 4 เส้นทาง และในการเขียนการทำงานพร้อมกัน ณ จุดใดจุดหนึ่ง สามารถกระตุ้นให้เกิดการทำงานพร้อมกันได้สูงสุด 4 เส้นทาง

7. ขนาดชาร์ตที่สามารถเขียนได้ มีขนาด 40 แถว x 17 คอลัมน์ โดยภายในชาร์ตไม่จำกัดจำนวนองค์ประกอบใดๆ และใน 1 โปรแกรมที่พัฒนา จะมีกี่ชาร์ตก็ได้ (แต่มีข้อแม้ว่าแต่ละชาร์ตห้ามใช้เอาต์พุตหมายเลข

เดียวกัน) จึงสามารถจะพัฒนาโปรแกรมที่มีขนาดใหญ่ได้ไม่จำกัด ทั้งนี้เมื่อแปลแล้วจะต้องไม่ใหญ่กว่าขนาดหน่วยความจำของ PC

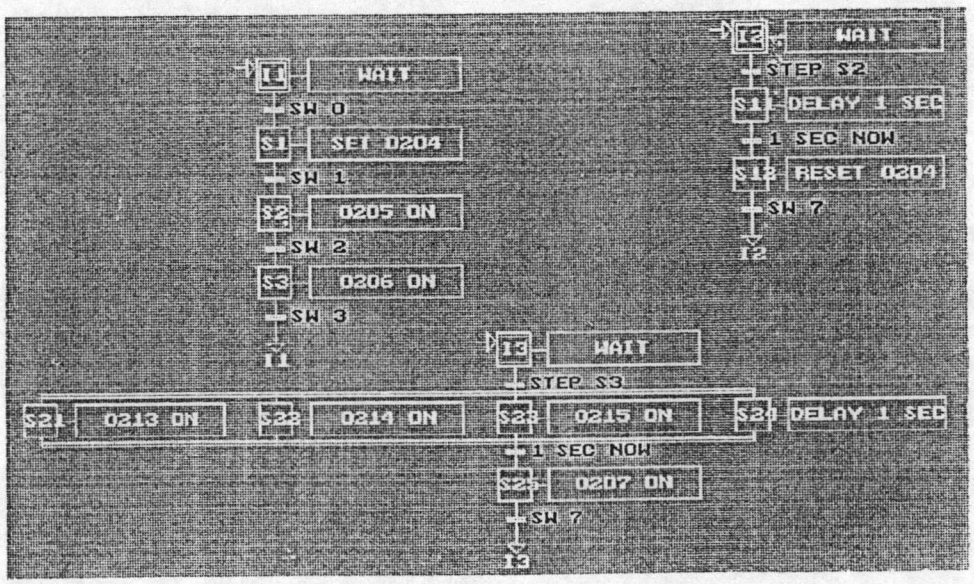
8. โปรแกรมนี้จะไม่มี Macro step แต่สามารถเขียนชาร์ต ให้มีลักษณะการทำงาน เช่นเดียวกับการใช้ Macro step ได้ ซึ่งจะกล่าวถึงในหัวข้อถัดไป

ซอฟต์แวร์ที่พัฒนาขึ้นมา จะประกอบด้วยโปรแกรม 2 โปรแกรม ได้แก่ FC.EXE และ FCMAIN.OVR โดย FCMAIN.OVR เป็นโปรแกรมส่วนโอเวอร์เลย์ (Overlay) ของ FC.EXE การพัฒนาโปรแกรมให้เป็นแบบโอเวอร์เลย์จะช่วยให้ในการใช้งาน ไม่ใช้หน่วยความจำ (RAM) ของระบบมาก โดยขณะทำงานจะมีเฉพาะเนื้อโปรแกรมที่ต้องใช้งานอยู่ในหน่วยความจำเท่านั้น และเมื่อใดที่ระบบต้องการใช้เนื้อโปรแกรมส่วนอื่นก็จะไปค้นจากโปรแกรมส่วนโอเวอร์เลย์ และอ่านขึ้นมาแทนที่ส่วนที่ไม่ได้ใช้งาน นอกจากนั้น ยังประกอบด้วยแฟ้ม "BRANDS.MNE" ซึ่งเก็บคำสั่งภาษานีมอนิคของ PC ต่างๆ ด้วย

9.1.2 <u>เทคนิคการเขียนชาร์ต</u>

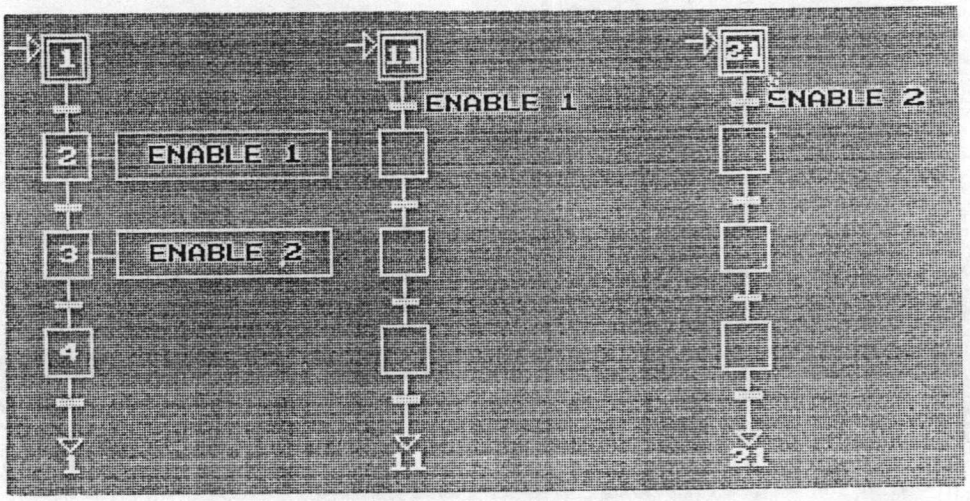
ในหัวข้อนี้จะกล่าวถึงเทคนิคการเขียนชาร์ตในรูปแบบต่างๆที่โปรแกรมที่พัฒนาขึ้นมาทำได้ตลอดจนชี้ให้เห็นจุดเด่นของภาษาฟังก์ชั่นชาร์ต

1. การเขียนโปรแกรมทำงานแบบพหุคูณ (Multitasking) : สามารถกระทำได้เนื่องจากข้อดีของ PC ที่สามารถรับโปรแกรมแบบพหุคูณ ได้ และเนื่องจากข้อดีของภาษาฟังก์ชั่นชาร์ตที่สามารถเขียนงานในลักษณะพหุคูณได้ วิธีการเขียนโปรแกรมแบบพหุคูณนี้ ทำได้โดยง่าย โดยการเขียนแยกแต่ละงานไว้บนชาร์ต และแต่ละงานจะต้องขึ้นต้นด้วยสเต็ปเริ่มต้นดังรูปที่ 9.1

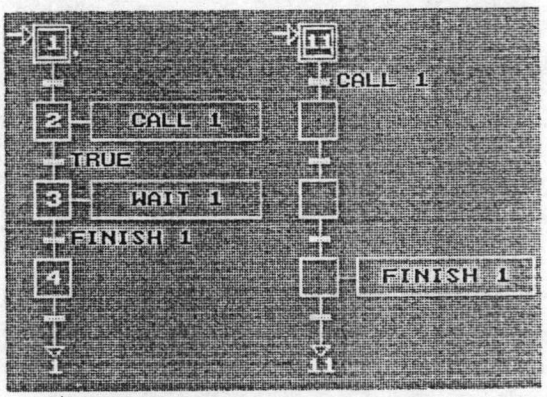


รูปที่ 9.1 : Multitasking

2. การออกแบบจากสูงสู่ต่ำ (Top-down design) : สามารถออกแบบได้ใน 2 รูปแบบ ได้แก่ แบบแรกเป็นการกระตุ้นให้เกิดการทำงาน มักจะใช้กับโปรแกรมหลักสำหรับควบคุมลำดับของงานต่างๆ ดังตัวอย่างในรูปที่ 9.2 แบบที่สองเป็นการเรียกใช้งานย่อย (Procedure) โดยจะเคลื่อนสเต็ปเมื่อทำงานย่อยนั้นเสร็จแล้ว ดังรูปที่ 9.3

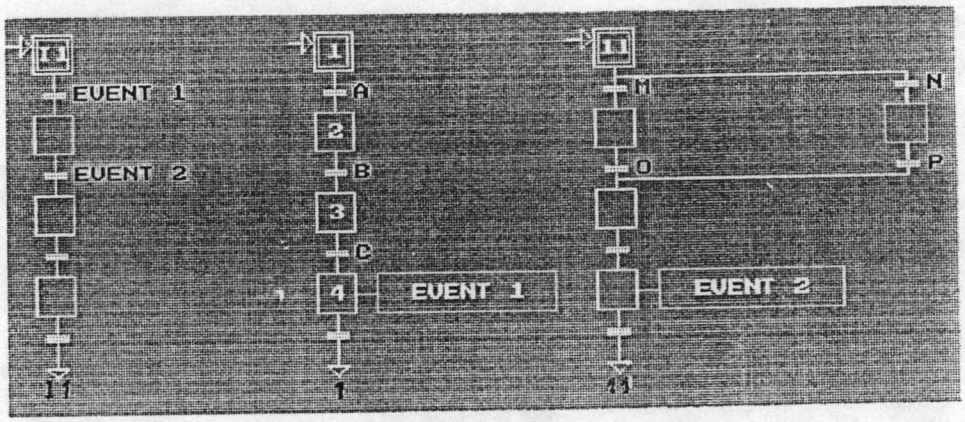


รูปที่ 9.2 : การควบคุมลำดับของงานย่อย



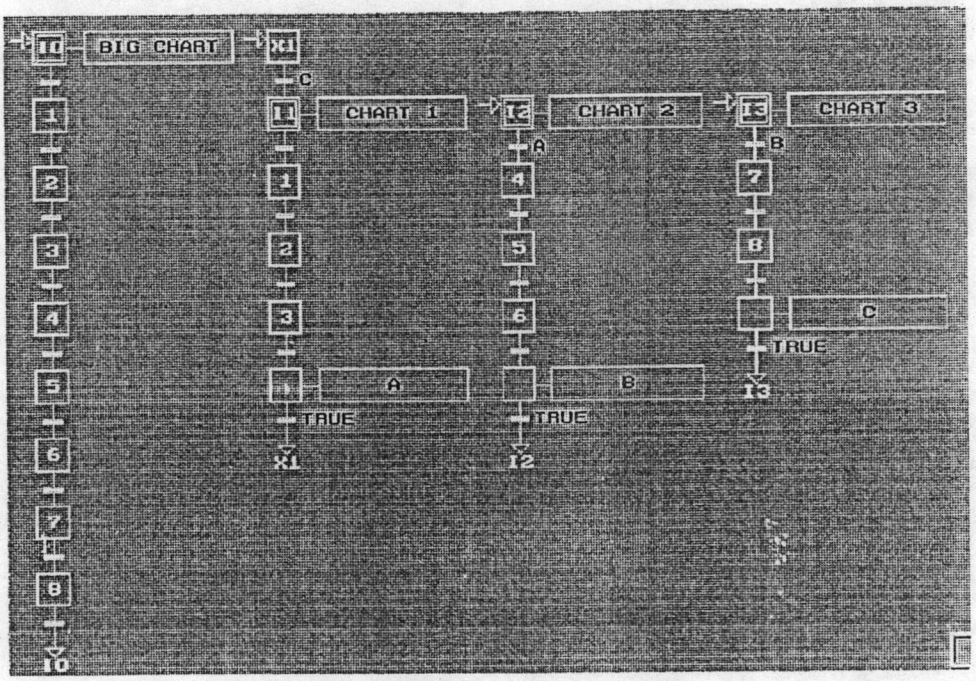
รูปที่ 9.3 : การเรียกใช้งานย่อย

3. เงื่อนไขเป็นลำดับเหตุการณ์ : จากรูปที่ 9.4 เงื่อนไขทรานสิชั่น T1 คือ EVENT1 ซึ่งเป็นผลมาจากลำดับการเกิดของสัญญาณจากหน้าสัมผัส A,B,C โดยเกิดสัญญาณ A และตามด้วย B และตามด้วย C ส่วนเงื่อนไขของทรานสิชั่น T2 คือ EVENT2 เป็นผลมาจาก ลำดับการเกิดสัญญาณจากหน้าสัมผัส M และ O หรือ N และ P



รูปที่ 9.4 : เงื่อนไขเป็นลำดับเหตุการณ์

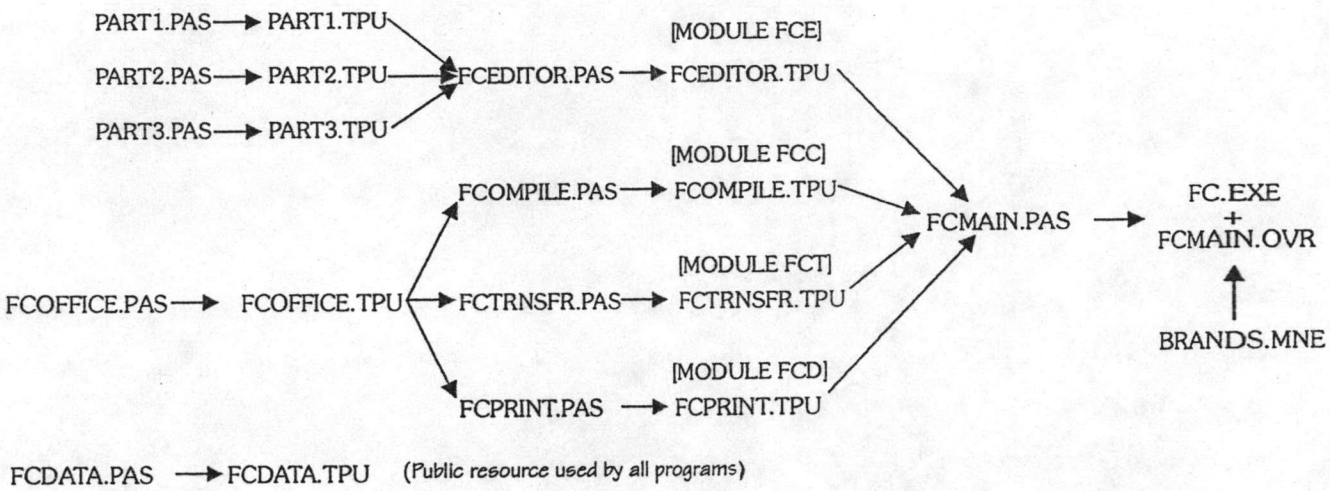
4. การเขียนโปรแกรมที่ใหญ่กว่า 1 ชาร์ต : จะต้องตัดโปรแกรมใหญ่นั้นแบ่งเขียนลงชาร์ตหลายชาร์ต โดยโปรแกรมแต่ละส่วนในแต่ละชาร์ตนั้น จะต้องจัดรูปใหม่ ตัวอย่างการเขียนพิจารณาจากรูปที่ 9.5



รูปที่ 9.5 : การเขียนโปรแกรมที่ใหญ่กว่า 1 ชาร์ต

9.2 สรุปการพัฒนา

ในการพัฒนาซอฟต์แวร์นี้ มีขั้นตอนการทำงานสรุปได้ดังแผนภาพ ในรูปที่ 9.6



รูปที่ 9.6 : ขั้นตอนการพัฒนา

ในการพัฒนาโมดูลตัววาดฟังก์ชั่นชาร์ต (FCE) ได้แก่ การพัฒนาโปรแกรม FCEDITOR.PAS โดยโปรแกรมนี้จะเรียกใช้งานยูนิตอื่นๆอีก 4 ยูนิต ได้แก่ FCDATA.TPU ซึ่งเป็นที่จัดเก็บทรัพยากรกลางที่ใช้ร่วมกันภายในระบบในทุกเรื่องๆ ได้แก่ ค่าคงที่ , ประเภทของตัวแปร , ตัวแปร , โปรแกรมย่อยต่างๆ เช่น การวาดภาพหน้าต่าง, การแสดงรายการคำสั่ง , การรับข้อความจากผู้ใช้ , การแสดงข่าวสารของระบบ , การย้ายมุมมองของจอภาพ เป็นต้น ยูนิตนี้แปลมาจาก FCDATA.PAS , PART1.TPU จัดการเกี่ยวกับรายละเอียดภายในเซลล์ ทั้งในเรื่องการรับและการแสดงผล และควบคุมการทำงานของเคอร์เซอร์ ยูนิตนี้แปลมาจาก PART1.PAS โดยอ้างอิงกับยูนิต FCDATA อีกทีหนึ่ง, PART2.TPU จัดการเรื่องความสัมพันธ์ขององค์ประกอบภายในชาร์ต แปลมาจาก PART2.PAS และอ้างอิงยูนิต FCDATA เช่นเดียวกัน , PART3.TPU จัดการเรื่องการวาดและการลบองค์ประกอบทุกๆองค์ประกอบ ในทุกๆเรื่อง เช่น การตรวจสอบกฎเกณฑ์, การวาดภาพ เป็นต้น ยูนิตนี้แปลมาจาก PART3.PAS โดยอ้างอิงกับยูนิต FCDATA ส่วนในโปรแกรม FCEDITOR เอง ก็จะทำหน้าที่ให้บริการต่อผู้ใช้ในเรื่องต่างๆ ตามรายการคำสั่งที่แสดง โดยจะไปเรียกใช้โปรแกรมย่อยอื่นๆอีกทีหนึ่ง เมื่อแปลโมดูลนี้แล้วจะได้โปรแกรม FCEDITOR.TPU

ในการพัฒนาโมดูลตัวแปลภาษาฟังก์ชั่นชาร์ต ก็จะเรียกใช้งาน ยูนิต FCDATA และยูนิต FCOFFICE โดยยูนิต FCOFFICE จะจัดการในเรื่องการอ่านข้อมูลจากแฟ้มขึ้นมาใช้งาน ส่วนงานอื่นๆในการแปลทั้งหมดจะอยู่ภายในโปรแกรม FCOMPILE.PAS ซึ่งเมื่อแปลแล้วจะได้โปรแกรม FCOMPILE.TPU โดยโปรแกรมนี้จะเรียกใช้แฟ้มข้อมูล "BRANDS.MNE" ในขณะทำงาน

การพัฒนาโมดูลตัวส่งโปรแกรม จะเป็นการพัฒนาโปรแกรม FCTRNSFR.PAS และการพัฒนาโมดูลตัวพิมพ์เอกสาร จะเป็นการพัฒนาโปรแกรม FCPRINT.PAS โดยทั้งสองโปรแกรมจะเรียกใช้ยูนิต FCDATA และ FCOFFICE เช่นเดียวกัน

เมื่อพัฒนาและทดสอบการทำงานของทั้ง 4 โมดูล จนเป็นที่น่าพอใจแล้ว จึงได้พัฒนาโปรแกรม FCMAIN.PAS ซึ่งจะมาทำหน้าที่เป็นตัวจัดการในการเรียกใช้โมดูลทั้ง 4 โดยเขียนโปรแกรมเป็นแบบโอเวอร์เลย์ ทำให้เมื่อแปลโปรแกรมนี้แล้วจะได้เป็น 2 โปรแกรม ได้แก่ FC.EXE และ FCMAIN.OVR

9.3 การใช้งาน

ก่อนอื่นจะต้องตรวจสอบว่ามีโปรแกรมที่ใช้งานอยู่ครบหรือไม่ ซึ่งได้แก่ FC.EXE, FCMAIN.OVR, BRANDS.MNE และ EGAVGA.BGI เมื่อต้องการใช้งานให้เรียกใช้โปรแกรม FC.EXE บนจอภาพก็จะปรากฏรายการคำสั่งหลักให้ผู้ใช้เลือกใช้ เมื่อต้องการวาดภาพชาร์ตให้เลือก "DRAW" โดยการกดปุ่ม F1 จอภาพก็จะแสดงหน้าต่างสำหรับวาดชาร์ตและด้านล่างจอภาพจะเป็นรายการคำสั่ง สำหรับการวาดภาพและงานอื่นที่เกี่ยวข้อง เมื่อต้องการจะออกจากการวาดภาพให้กดปุ่ม "ESC" จอภาพก็จะแสดงภาพรายการคำสั่งหลัก หากผู้ใช้ต้องการจะแปลชาร์ตที่ได้เขียนขึ้น ให้เรียกใช้คำสั่ง "COMPILE" โดยการกดปุ่ม F2 ระบบจะแสดงหน้าต่างชาร์ตและรายการคำสั่งสำหรับการแปล ในการแปลชาร์ตไปเป็นสมการบูลลีนให้กด F1 (BOOLEAN) ระบบจะแสดงข้อความให้ผู้ใช้ทราบว่ากำลังแปลเป็นสมการบูลลีน เมื่อแปลเสร็จก็จะแจ้งให้ผู้ใช้ทราบ ส่วนการแปลไปเป็นภาษานีมอนิคจะต้องเลือกก่อนว่าจะแปลเป็นภาษานีมอนิคของบริษัทใด โดยกดปุ่ม F3:BRAND และป้อนยี่ห้อของ PC ที่ต้องการ จากนั้นกดปุ่ม F2:MNE เพื่อแปลเป็นภาษานีมอนิค โดยระบบจะแปลเป็นภาษานีมอนิคตามที่ระบุและแปลเป็นภาษาเครื่องของออมร่อน เพื่อใช้ส่งไปยัง PC การออกจากการแปลทำได้โดยการกด "ESC" หากผู้ใช้ต้องการจะส่งโปรแกรมไปยัง PC ให้กด F3 ของรายการคำสั่งหลัก ระบบจะแสดงชุดคำสั่งในการส่งโปรแกรม ผู้ใช้สามารถระบุชาร์ตที่ต้องการส่งไปยัง PC ได้โดยกด F5:LOAD และเมื่อต้องการส่งให้กด F3 ระบบก็จะแสดงข้อความว่ากำลังส่งโปรแกรม ขณะส่งโปรแกรมทุกๆ 2 กิโลไบต์ ระบบจะส่งเสียงสั้นๆบอก ในการทำเอกสารให้ผู้ใช้เลือกใช้คำสั่ง "DOCUMENT" ที่รายการคำสั่งหลัก ระบบจะแสดงชุดคำสั่งสำหรับสั่งพิมพ์เอกสารต่างๆ ผู้ใช้สามารถเลือกพิมพ์เอกสารได้ตามต้องการ ในการเลิกใช้งานโปรแกรมนี้ให้ผู้ใช้กด ESC ที่รายการคำสั่งหลัก

9.4 ข้อเสนอแนะ

อาจกล่าวได้ว่าซอฟต์แวร์ที่พัฒนาขึ้นนี้ เป็นเพียงงานส่วนหนึ่งในการพัฒนาโปรแกรมสำหรับพัฒนาภาษาฟังก์ชั่นชาร์ต บน PC ซอฟต์แวร์ที่สมบูรณ์ นอกจากจะมีความสามารถดังซอฟต์แวร์ที่พัฒนาแล้ว คือ เขียนโปรแกรมได้, แปลได้, ส่งโปรแกรมได้ และทำเอกสารได้ ยังควรจะจะสามารถใช้ในการค้นหาจุดบกพร่อง (Debug) ของโปรแกรมได้ และสามารถจะใช้จำลองการทำงานของโปรแกรม (Simulation) ได้อีกด้วย ก็จะทำให้เป็นชุดพัฒนาโปรแกรมภาษาฟังก์ชั่นชาร์ตที่สมบูรณ์แบบ