

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องมือช่วยพัฒนาโปรแกรมสำหรับไมโครคอนโทรลเลอร์

Tools for Developing Programs on Microcontroller



T117321



นางสาวใจแก้ว จิราวัฒน์พงษ์

นายณัฐพล บุญดา

เลขที่ 117321  
ลงทะเบียน  
วันเดือนปี 20 ก.ค. 2554

12339325  
b.....  
i.....

โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานี้เท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาหรือข้อมูลซึ่งอาจถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปีการศึกษา 2553

**Tools for Developing Programs on Microcontroller**



**MS. JAIKAEW JIRAWATTANAPONG**  
**MR. NATAPON BOONDA**

**A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIRMENT FOR THE DEGREE OF BACHELOR OF SCIENCE  
IN COMPUTER SCIENCE  
FACULTY OF SCIENCE**

**KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

**ACADEMIC YEAR 2010**

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงการพิเศษ เครื่องมือช่วยพัฒนาโปรแกรมสำหรับไมโครคอนโทรลเลอร์  
Tools for Developing Programs on Microcontroller

ชื่อนักศึกษา นางสาวใจแก้ว จิราวัฒน์พงษ์ 50050110  
นายณัฐพล บุญดา 50050126

ปริญญา วิทยาศาสตรบัณฑิต  
สาขาวิชา วิทยาการคอมพิวเตอร์  
อาจารย์ที่ปรึกษา รศ. ชีรวัฒน์ ประกอบผล

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้  
โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการ  
คอมพิวเตอร์ ประจำปีการศึกษา 2553

คณะกรรมการสอบ	ลายมือชื่อ
ผศ.ดร.จิรพร วีระพันธุ์ ประธานกรรมการ	
ดร.รุ่งรัตน์ เวียงศรีพนาวัลย์ กรรมการ	
รศ.ชีรวัฒน์ ประกอบผล กรรมการและอาจารย์ที่ปรึกษา	

### ลิขสิทธิ์ของคณะวิทยาศาสตร์

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง การใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงการพิเศษ	เครื่องมือช่วยพัฒนาโปรแกรมสำหรับไมโครคอนโทรลเลอร์
ชื่อนักศึกษา	นางสาวใจแก้ว จิราวัฒน์พงษ์ นายณัฐพล บุญคา
ปริญญา	วิทยาศาสตรบัณฑิต
สาขาวิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2553
อาจารย์ที่ปรึกษา	รศ. ชีรวัฒน์ ประกอบผล

### บทคัดย่อ

โครงการพิเศษฉบับนี้นำเสนอถึงเครื่องมือช่วยพัฒนาโปรแกรมสำหรับไมโครคอนโทรลเลอร์ ซึ่งช่วยในการอำนวยความสะดวกแก่ผู้เขียนโปรแกรมไมโครคอนโทรลเลอร์ตระกูล MCS-51 ซีพียู MPC82G516A โดยผู้ใช้สามารถวาดโฟลว์ชาร์ตแบบวิซวล และระบุข้อกำหนดเกี่ยวกับไมโครคอนโทรลเลอร์ที่ต้องการนำโปรแกรมไปรันโฟลว์ชาร์ตได้ ซึ่งจะได้โค้ดเป็นภาษาซีสำหรับไมโครคอนโทรลเลอร์ตามที่ใช้ระบุอย่างอัตโนมัติ โดยที่ผู้พัฒนาโปรแกรมไม่ต้องเขียนโปรแกรมด้วยตัวเอง โค้ดที่สร้างขึ้นมานั้นจะสามารถคอมไพล์ได้ด้วยคอมไพเลอร์ของเคิล ซึ่งเมื่อคอมไพล์ออกมาแล้วจะได้เป็นไฟล์นามสกุล .hex หรือเลขฐาน 16 จากนั้นนักพัฒนาสามารถนำโค้ดที่ได้นั้นไปพัฒนาเพิ่มเติมและนำไปคอมไพล์เพื่อไปรันบนอุปกรณ์จริงต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<b>Title</b>	Tools for Developing Programs on Microcontroller
<b>Students</b>	Ms.Jaikaew Jirawattanapong Mr.Nattapon Boonda
<b>Degree</b>	Bechelor of Science
<b>Major Program</b>	Computer Science
<b>Academic Year</b>	2010
<b>Advisor</b>	Assoc.Prof. Teerawat Prakobphon

### ABSTRACT

This thesis presents Tools for Developing Programs on Microcontroller which Facilitate the programming of microcontroller MCS-51 of chip number MPC82G516A. The user can draw flowchart and specific requirements to the microcontroller and program will automatically generated in C language. This code can compile by Keil which export in .hex or hexa number then the developer can take this code to additional develop and be compiled to run on the device further.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

โครงการพิเศษเรื่องเครื่องมือช่วยพัฒนาโปรแกรมสำหรับไมโครคอนโทรลเลอร์สำเร็จรูป - ถ่วงได้นั้นเนื่องจากได้รับความกรุณาจากรศ.ธีรวัฒน์ ประกอบผล ซึ่งทำหน้าที่กรรมการและอาจารย์ที่ปรึกษา ได้ให้คำแนะนำและคำปรึกษาที่ดีในทุกๆด้านมาโดยตลอดทำให้โครงการพิเศษนี้เสร็จสมบูรณ์ ขอบพระคุณ ผศ.ดร.จิรพร วีระพันธุ์ ซึ่งเป็นประธานกรรมการและ ดร.รุ่งรัตน์ เวียงศรีพนาวัลย์ ซึ่งเป็นกรรมการสอบที่ได้ให้คำแนะนำในการปรับปรุงโครงการพิเศษให้สมบูรณ์ยิ่งขึ้น และขอขอบคุณเพื่อนๆที่คอยช่วยเหลือ ให้คำปรึกษาและให้กำลังใจมาโดยตลอด

สุดท้ายนี้ขอขอบคุณครอบครัวที่ให้การสนับสนุนทางด้านการศึกษาและให้คำปรึกษาในยามมีปัญหาจนกระทั่งสำเร็จการศึกษาในระดับปริญญาตรี

นางสาวใจแก้ว จิราวัฒนพงษ์  
นายณัฐพล บุญคา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VI
สารบัญรูป	VII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของงานวิจัย	1
1.2 วัตถุประสงค์ของงานวิจัย	1
1.3 ขอบเขตของงานวิจัย	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง	3
2.1 ทฤษฎีและความรู้เกี่ยวกับไมโครคอนโทรลเลอร์ตระกูล MCS-51	3
2.2 ภาพรวมของ MPC82G516A	4
2.3 ส่วนประกอบของ MPC82G516A	5
2.4 การทำงานของ MPC82G516A	6
2.5 รูปแบบตัวถังและการขา	8
2.6 การสร้างโค้ด	14
บทที่ 3 ขั้นตอนและการออกแบบระบบ	18
3.1 ขั้นตอนการดำเนินงาน	18
3.2 Class Diagram	28
บทที่ 4 ผลการทดลอง	35
4.1 สถาปัตยกรรมของระบบและซอฟต์แวร์	35
4.2 ลักษณะของ Pop up Menu ของ Model ต่างๆ	38
4.3 Property ที่สำคัญของแต่ละ Component ของโคแอะแกรม	46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้า
บทที่ 5 ผลการดำเนินงาน ข้อจำกัด และข้อเสนอแนะ	48
5.1 ผลการดำเนินงาน	48
5.4 ข้อจำกัด	48
5.3 ข้อเสนอแนะ	49
เอกสารอ้างอิง	50
ภาคผนวก ก    คู่มือการติดตั้งโปรแกรม	51
ภาคผนวก ข    คู่มือการใช้งานโปรแกรม	55
ภาคผนวก ค    ตัวอย่างขั้นตอนการสร้างโปรแกรม	74



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง

ตารางที่	หน้า
2.1 รายละเอียดของขาต่างๆ	8
2.2 การ Parse แบบ Shift-Reduce	15
2.3 Parsing ของ Grammar expression	16
2.4 การ Move ของ LR Parser บน $id * id + id$	17
3.1 ตารางการเก็บข้อมูลจากโฟลว์ชาร์ต	25
3.2 ตารางการเก็บข้อมูลหลังจากที่มีการจัดเรียงข้อมูลในตารางที่ 3.1	25



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญภาพ

ภาพที่	หน้า
2.1 บล็อกไคอะแกรมแสดงการทำงานของ MPC82G516A	7
2.2 แสดงการทำงานของ PLCC 44 ขา	8
3.1 ส่วนประกอบต่างๆบนบอร์ดทดลอง	18
3.2 อัลกอริทึมการตรวจสอบ Syntax ของผังงานโดยรวม	21
3.3 ตัวอย่างการวาดโฟลว์ชาร์ต	24
3.4 ตัวอย่างบางส่วนของโฟลว์ชาร์ต	26
3.5 คลาสหลัก Visual_Programming	29
3.6 คลาส MainFrame	30
3.7 คลาส SymbolTable	31
3.8 คลาส StartActionListener	32
3.9 คลาส CheckModels	33
3.10 คลาส SaveDiagramInFile	34
4.1 GUI ของโปรแกรม Visual Programming สำหรับการสร้างโค้ด	36
4.2 Pop up Menu ของ Start Model	38
4.3 Pop up การกำหนดค่าทางซ้ายของ Assignment Model	39
4.4 Pop up Menu การกำหนดค่าทางด้านขวาของ Assignment Model	39
4.5 การกำหนดค่าการกระทำทางคณิตศาสตร์ของ Assignment Model	40
4.6 Operator ของการกำหนดค่าการกระทำทางคณิตศาสตร์	40
4.7 Pop up Menu ของ Input Model	41
4.8 Pop up Menu ของ Output Model	41
4.9 Pop up Menu ของ Condition Model โดยการกำหนดค่าทางด้านซ้าย	42
4.10 Operator ของ Condition Model	43
4.11 Pop up Menu ของ For Model	43
4.12 Pop up Menu ของ Buzzer Model	44
4.13 Pop up Menu ของ LED Model	44
4.14 Pop up Menu ของ Delay Model	45
4.15 Pop up Menu ของ Number Model	45
4.16 Pop up Menu ของ Relay Model	46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับอาจารย์ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญภาพ (ต่อ)

ภาพที่	หน้า
ก.1 หน้าจอโปรแกรมติดตั้งแสดงทางเลือกของภาษา	51
ก.2 หน้าจอโปรแกรมติดตั้งแสดงข้อความต้อนรับ	52
ก.3 หน้าจอโปรแกรมติดตั้งแสดงที่เก็บโปรแกรม	52
ก.4 หน้าจอโปรแกรมติดตั้งแสดงกลุ่มของโปรแกรม	53
ก.5 หน้าจอโปรแกรมติดตั้งแสดงตัวเลือกเกี่ยวกับไอคอน	53
ก.6 หน้าจอโปรแกรมติดตั้งแสดงข้อความการติดตั้งเสร็จสมบูรณ์	54
ก.7 หน้าจอแสดงผล Visual Programming	54
ข.1 หน้าจอเริ่มต้นโปรแกรม Visual Programming	55
ข.2 หน้าจอแรกเมื่อมีการเปิด New Project ขึ้นมา	56
ข.3 การวาดผังงาน	57
ข.4 การวาดเส้นของผังงาน	58
ข.5 การระบุค่าของโมเดล Start	59
ข.6 การแสดงข้อความผิดพลาดเมื่อผู้ใช้กรอกข้อมูลของผังงานไม่ถูกต้อง	59
ข.7 Pop up Menu ของการลบตัวแปร	60
ข.8 Pop up Menu ของการกำหนดค่าทางด้านซ้ายของ Expression	60
ข.9 Pop up Menu ของการกำหนดค่าทางขวาของ Expression	61
ข.10 Pop up Menu ที่ใช้กำหนดค่าของการกระทำทางคณิตศาสตร์	62
ข.11 Pop up Menu ที่ทำการกำหนดค่าของโมเดล Input	62
ข.12 Pop up Menu ที่ทำการกำหนดค่าของโมเดล Output	63
ข.13 Pop up Menu ที่ทำการกำหนดค่าของโมเดล Condition	63
ข.14 Pop up Menu ที่ทำการกำหนดค่าของโมเดล While	64
ข.15 Pop up Menu ที่ทำการกำหนดค่าของโมเดล EndDoWhile	64
ข.16 Pop up Menu ที่ทำการกำหนดค่าของโมเดล For	65
ข.17 Pop up Menu ที่ทำการกำหนดค่าของโมเดล Buzzer	65
ข.18 Pop up Menu ที่ทำการกำหนดค่าของโมเดล LED	66
ข.19 Pop up Menu ที่ทำการกำหนดค่าของโมเดล Delay	66
ข.20 Pop up Menu ที่ทำการกำหนดค่าของโมเดล Number	67
ข.21 Error Dialog กรณีที่ไม่มีโมเดล Stop ในไฟล์วาร์ด	68

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ในการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญภาพ (ต่อ)

ภาพที่	หน้า
ข.22 Error Dialog กรณีที่ไม่มีการลากเส้นเชื่อมใดๆ ระหว่างโมเดล	68
ข.23 กรณีที่มีการลากเส้นเชื่อมระหว่างโมเดลไม่ครบทุกโมเดลและโมเดล Condition มีเส้นออกไม่ครบ 2 เส้น	69
ข.24 Error Dialog กรณีที่โมเดล Input ไม่มีโค้ด	69
ข.25 Error Dialog กรณีที่โมเดล While ไม่สมบูรณ์	69
ข.26 โฟลว์ชาร์ตที่มีการวางโมเดลไม่ถูกตำแหน่ง	70
ข.27 Error Dialog กรณีที่1 โมเดล For วางผิดตำแหน่ง	70
ข.28 หน้าจอหลังจากการกดเมนู generate code	71
ข.29 หน้าจอเมื่อมีการกดเมนู file >> Save Diagram	72
ข.30 หน้าจอของโปรแกรมเมื่อมีการเรียกใช้ฟังก์ชันมาใช้ใหม่	73
ค.1 Pop up Menu ของโมเดล Start ตัวอย่างที่ 1	75
ค.2 Pop up Menu ของโมเดล Delay ตัวอย่างที่ 1	75
ค.3 Pop up Menu ของโมเดล Number ตัวอย่างที่ 1	76
ค.4 Pop up Menu ของโมเดล LED ตัวอย่างที่ 1	77
ค.5 Pop up Menu ของโมเดล Buzzer ตัวอย่างที่ 1	78
ค.6 Pop up Menu ของโมเดล Delay ที่ 2 ตัวอย่างที่ 1	78
ค.7 Pop up Menu ของโมเดล Number ที่ 2 ตัวอย่างที่ 1	79
ค.8 Pop up Menu ของโมเดล LED ที่ 2 ตัวอย่างที่ 1	79
ค.9 Pop up Menu ของโมเดล Buzzer ที่ 2 ตัวอย่างที่ 1	80
ค.10 Pop up Menu ของโมเดล Delay ที่ 3 ตัวอย่างที่ 1	80
ค.11 Pop up Menu ของโมเดล Delay ที่ 3 ตัวอย่างที่ 1	81
ค.12 แผนผังโฟลว์ชาร์ตตัวอย่างที่ 1	81
ค.13 โค้ดที่ได้หลังการ generate ของตัวอย่างโปรแกรมที่ 1	82
ค.14 ภาพรวมของตัวอย่างโปรแกรมที่ 1	82
ค.15 Pop up Menu ของโมเดล Start ตัวอย่างที่ 2	83
ค.16 Pop up Menu ของโมเดล For ตัวอย่างที่ 2	84
ค.17 Pop up Menu ของโมเดล Delay ตัวอย่างที่ 1	84
ค.18 Pop up Menu ของโมเดล Condition ตัวอย่างที่ 2	86

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เฉพาะอาจารย์ประจำภาคนั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญภาพ (ต่อ)

ภาพที่	หน้า
ค.19 Pop up Menu ของ โมเดล Relay ตัวอย่างที่ 2	87
ค.20 Pop up Menu ของ โมเดล LED ตัวอย่างที่ 2	88
ค.21 Pop up Menu ของ โมเดล Relay ที่ 2 ตัวอย่างที่ 2	88
ค.22 Pop up Menu ของ โมเดล LED ที่ 2 ตัวอย่างที่ 2	89
ค.23 แผนผังไฟลิวชาร์ตตัวอย่างที่ 2	90
ค.24 โค้ดที่ได้หลังการ generate ของตัวอย่างโปรแกรมที่ 2	91
ค.25 ภาพรวมของตัวอย่างโปรแกรมที่ 2	91



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 1

### บทนำ

#### 1.1 ความเป็นมาและความสำคัญ

ปัจจุบันงานทางด้านไมโครคอนโทรลเลอร์มีบทบาทในชีวิตประจำวันมากขึ้น โดยเฉพาะในระบบควบคุมของอุปกรณ์ไฟฟ้าต่างๆ และการเขียนโปรแกรมให้ระบบทำงานนั้นแต่เดิมจะใช้ภาษาแอสเซมบลีในการพัฒนาโปรแกรมเป็นหลัก แต่สำหรับการพัฒนาระบบที่มีความซับซ้อนมากขึ้นมี การทำงานทางคณิตศาสตร์มาก การใช้ภาษาแอสเซมบลีในการพัฒนาโปรแกรมจะเขียนโปรแกรมได้ลำบาก ต้องใช้เวลามากและยังทดสอบได้ยากด้วย จึงมีการนำภาษาระดับสูงมาช่วยในการพัฒนางานทางด้านไมโครคอนโทรลเลอร์ โดยเมื่อเขียนโปรแกรมต่างๆออกมาแล้ว จะต้องใช้โปรแกรมแปลภาษาระดับสูงให้เป็นภาษาแอสเซมบลี หรือรหัสภาษาเครื่องที่ไมโครคอนโทรลเลอร์เข้าใจอีกครั้งหนึ่ง จากนั้นจึงนำรหัสที่ได้ไปโปรแกรมให้กับตัวไมโครคอนโทรลเลอร์เพื่อให้ระบบทำงาน ภาษา C เป็นอีกภาษาหนึ่งที่ถูกนำมาใช้ในการพัฒนางานด้านไมโครคอนโทรลเลอร์ ซึ่งเป็นภาษาที่มีโครงสร้างทางภาษาที่ดีและความเร็วใกล้เคียงกับภาษาแอสเซมบลี นอกจากนี้การใช้ภาษาระดับสูงยังสามารถช่วยในการเขียนโปรแกรมให้ไมโครคอนโทรลเลอร์ทำงานทางคณิตศาสตร์ได้ง่ายขึ้นอีกด้วย

ด้วยเหตุผลนี้เอง จึงทำให้เกิดแรงจูงใจในการสร้างโปรแกรมที่มีส่วนช่วยในการสร้างโค้ดภาษา C สำหรับไมโครคอนโทรลเลอร์แบบอัตโนมัติ (Automatic Code generation) ทำให้การสร้างโปรแกรมควบคุมระบบไมโครคอนโทรลเลอร์ง่ายขึ้น และลดความผิดพลาดจากการสร้างโปรแกรมด้วยมือ

#### 1.2 วัตถุประสงค์ของโครงงานวิจัย

1.2.1 เพื่อสร้างโปรแกรมอำนวยความสะดวกในการเขียนโปรแกรมให้กับระบบไมโครคอนโทรลเลอร์โดยผู้ใช้งานไม่จำเป็นต้องเขียนโปรแกรมด้วยตนเอง โปรแกรมนี้จะช่วยในการสร้างโค้ดออกมาเป็นภาษา C สำหรับไมโครคอนโทรลเลอร์ได้อัตโนมัติ

1.2.2 เพื่อศึกษาในส่วนของทฤษฎี LR Parsing และอัลกอริทึมของการสร้างโค้ดซึ่งมีความสำคัญในการนำไปสร้างโค้ดแบบอัตโนมัติ สำหรับงานระบบไมโครคอนโทรลเลอร์ด้วยภาษา C

1.2.3 เพื่อศึกษาความเป็นไปได้ในการปรับเครื่องมือเพื่อการสร้างโค้ดสำหรับไมโครคอนโทรลเลอร์ตระกูลอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 1.3 ขอบเขตของโครงการงาน

1.3.1 สร้าง Menu Bar, Tool Bar และ User Interface ต่างๆ ที่เกี่ยวข้องกับงานทางด้านไมโครคอนโทรลเลอร์ในส่วนของการสร้างฟิวเจอร์บอร์ด ยกตัวอย่างเช่น

- ส่วนของพอร์ต ได้แก่ อินพุตพอร์ต สามารถรับอินพุตพอร์ตแบบบิตได้, เอาท์พุตพอร์ต สามารถ เลือกเป็นแบบบิต ไบต์ หรือเป็นตัวแปรได้ เป็นต้น
- ส่วนของการหน่วงเวลา (Delay)
- ส่วนของรุ่นของไมโครคอนโทรลเลอร์ โดยใช้ตระกูล MCS-51
- ส่วนของฟังก์ชันเฉพาะสำหรับบอร์ดทดลอง เช่น DISNUM FUNCTION, BUZZER FUNTION เป็นต้น

1.3.2 เพิ่มเติมส่วนตัวดำเนินการต่างๆ ในส่วนของการคำนวณคณิตศาสตร์และตรรกะของโปรแกรม (Arithmetic Expression) เพื่อให้เหมาะสมกับการออกแบบโปรแกรมสำหรับไมโครคอนโทรลเลอร์ ยกตัวอย่าง เช่น and , or และ xor เป็นต้น

1.3.3 ปรับปรุง Pop Up ในแต่ละ Model ในองค์ประกอบของฟิวเจอร์บอร์ดที่มีอยู่ให้เหมาะสมกับไมโครคอนโทรลเลอร์ ยกตัวอย่างเช่น ในส่วนของ Input Port, Output Port และ Assign ที่ใช้ในการรับส่งข้อมูล เพื่อเป็นข้อมูลในการสร้างโค้ดต่อไป

### 1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 ความรู้เกี่ยวกับระบบการทำงานของไมโครคอนโทรลเลอร์

1.4.2 ซอฟต์แวร์เครื่องมือเพื่อช่วยในการพัฒนาโปรแกรม โดยมีลักษณะการทำงานติดต่อกับผู้ใช้แบบ Graphic User Interface สำหรับไมโครคอนโทรลเลอร์ และสามารถสร้างชุดคำสั่งภาษา C สำหรับไมโครคอนโทรลเลอร์ได้อย่างอัตโนมัติ และสามารถนำไปใช้งานได้งานจริงกับบอร์ดทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

# ทฤษฎีและความรู้ที่เกี่ยวข้อง

ในการจัดทำโครงการพิเศษนี้ได้ใช้ทฤษฎีและความรู้ที่เกี่ยวข้อง ดังนี้

- ทฤษฎีและความรู้เกี่ยวกับไมโครคอนโทรลเลอร์ตระกูล MCS-51
- การสร้างโค้ด

### 2.1 ทฤษฎีและความรู้เกี่ยวกับไมโครคอนโทรลเลอร์ตระกูล MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 หรือ 8051 เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต มีผลิตรายหลายบริษัทแต่ละรุ่นมีประสิทธิภาพที่แตกต่างกัน เบอร์ที่เป็นพื้นฐานคือเบอร์ 8051 มีขา 40 ขา เบอร์ 89C51 เป็นไอซีที่มีโครงสร้างเป็น CMOS ที่ประหยัดพลังงาน เบอร์ 8751 เป็นเบอร์ที่มี EPROM สำหรับเก็บโปรแกรมอยู่ในแต่การลบข้อมูลเก่าออกจะต้องแสงอัลตราไวโอเลตในการลบ สำหรับรุ่น 8052 จะมีหน่วยความจำโปรแกรมขนาดใหญ่ขึ้น มีตัวนับตัวจับเวลาให้ใช้งานมากกว่าเบอร์อื่นๆ ส่วนเบอร์ AT89C1051 และ AT89C2051 ของบริษัท Atmel เป็นตระกูล MCS-51 ที่มีขา 20 ขา เหมาะสำหรับงานเล็กๆ ต้องการใช้พอร์ตไม่มาก และใช้พลังงานไฟฟ้าน้อยกว่ารุ่นที่มีขา 40 ขา สำหรับปัจจุบันนิยมใช้รุ่นที่มีหน่วยความจำแบบ Flash สำหรับเก็บโปรแกรมอยู่ใน

การเขียนโปรแกรมให้กับไมโครคอนโทรลเลอร์นั้นยุคแรกๆ จะต้องเขียนด้วยภาษาแอสเซมบลี จากนั้นแปลโปรแกรมที่ได้ให้เป็นรหัสภาษาเครื่องแล้วจึงโหลดลงหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ ต่อมาได้มีตัวแปลภาษา C สำหรับไมโครคอนโทรลเลอร์เบอร์ต่างๆ ทำให้ผู้พัฒนาสามารถเขียนโปรแกรมด้วยภาษา C ได้ จากนั้นจึงแปลให้เป็นรหัสภาษาเครื่องแล้วจึงโหลดลงสู่หน่วยความจำ เมื่อเขียนด้วยภาษา C ได้ทำให้การพัฒนาทางด้านสมองกลฝังตัวทำได้ง่ายขึ้น แต่ในยุคแรกๆ การโปรแกรมลงสู่หน่วยความจำทำได้ยาก เนื่องจากต้องใช้เครื่องมือเฉพาะในการโปรแกรม แต่ปัจจุบันสามารถเขียนโปรแกรมและโหลดลงสู่ตัวชิปได้ง่ายขึ้น เนื่องจากชิปรุ่นใหม่ๆ จะมีโปรแกรมมอนิเตอร์สำหรับสื่อสารข้อมูลแบบอนุกรมกับคอมพิวเตอร์อยู่ในตัวชิป ไมโครคอนโทรลเลอร์ของบริษัท Megawin เป็นไมโครคอนโทรลเลอร์ตระกูล MCS-51

ในบทนี้จะกล่าวถึง โครงสร้างและการทำงานพื้นฐานของไมโครคอนโทรลเลอร์ โดยจะเน้นไปที่ชิปเบอร์ MPC82G516A ของบริษัท Megawin ที่นำมาใช้ในการสร้างชุดทดลองในโครงการนี้ ซึ่งเป็นชิปที่มีราคาไม่แพงและสามารถเขียนโปรแกรมเข้าไปในตัวชิปได้โดยตรง โดยจะเริ่มอธิบายถึง ภาพรวมของ MPC82G516A ส่วนประกอบหลักๆ ซึ่งมีโครงสร้างภายในคล้ายกับ MCS-51 มาตรฐาน แต่จะมีคุณสมบัติที่เพิ่มเติมขึ้นมา เช่น มีหน่วยความจำภายในแบบแฟลช การ

โปรแกรมแบบ ISP(In-System Programming) และสุดท้ายจะอธิบายถึงการทำงานพื้นฐานของ MPC82G516A

## 2.2 ภาพรวมของ MPC28G516A

MPC82G516A เป็นไมโครคอนโทรลเลอร์ชิปเดี่ยว สามารถประมวลผลคำสั่งได้ใน 1-7 รอบสัญญาณนาฬิกา และมีชุดคำสั่งที่ทำงานเหมือนกับ 8051 เพราะฉะนั้นจึงมีลักษณะต่างๆที่เป็นมาตรฐานเหมือน 8051 แต่ชิป MPC82G516A สามารถทำงานได้ด้วยความเร็วต่ำได้ ใช้แรงดันไฟฟ้าต่ำว่าได้ และด้วยเหตุนี้พลังงานที่ถูกใช้ก็จะลดน้อยลงไปด้วย

MPC82G516A มีหน่วยความจำภายในแบบแฟลช(Flash)ขนาด 64 กิโลไบต์ สำหรับเก็บโปรแกรมและข้อมูล โดยโปรแกรมได้แบบ In-system Programming(ISP) ซึ่งบนชิปจะมีโปรแกรม Boot Loader บรรจุอยู่มาตั้งแต่การผลิตมาจากโรงงาน และแบบ In-Circuit Programming(ICP) สามารถที่จะทำงานได้ทั้งแบบขนาน (parallel) และอนุกรม(serial) และยังสามารถในการโปรแกรมลงบนไมโครคอนโทรลเลอร์แบบ In-Application Programming(IAP) ISP และ ICP อนุญาตให้ผู้ใช้สามารถเขียนหรือลบโปรแกรมได้โดยตรง โดยไม่ต้องถอดชิปออกจากแผงวงจร สำหรับ IAP หมายถึง ความสามารถในการโปรแกรมลงในหน่วยความจำภายในแบบแฟลช ขณะที่โปรแกรมกำลังทำงานอยู่

ส่วนประกอบหลักของ MPC82G516A ได้แก่ หน่วยความจำ scratch-pad RAM ขนาด 256 ไบต์ พอร์ต I/O ขนาด 8 บิต 4 พอร์ต ตัวจับเวลา/ตัวนับ 3 ตัว พอร์ตอนุกรมซึ่งสามารถรับส่งข้อมูลแบบฟลูอิดเพลิกซ์ วงจรอินเทอร์รัปต์ นอกจากนี้ยังมีหน่วยความจำแรมเพิ่มเติม(XRAM) ขนาด 1024 ไบต์ พอร์ต I/O แบบพิเศษ(P4) ตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอล(ADC) ขนาด 10 บิต โปรแกรมวงจรถ่ายภาพ(PCAP) การสื่อสารแบบอนุกรม(SPI) การเชื่อมต่อผ่านพอร์ตอนุกรมกับคอมพิวเตอร์(UART) keypad interrupt วอตช์ดอกไทเมอร์(Watchdog Timer) และอื่นๆอีก นอกจากนี้ MPC82G516A ยังกลายเป็นไมโครคอนโทรลเลอร์ที่มีประสิทธิภาพ เหมาะสำหรับ แอปพลิเคชันที่หลากหลาย

MPC82G516A มีโหมดประหยัดพลังงาน 2 โหมด และการกำหนดระบบสัญญาณนาฬิกา เพื่อลดการใช้พลังงาน ใน Idle mode CPU จะไม่ทำงาน ในขณะที่การทำงานของอุปกรณ์ต่อพ่วงอื่นๆและการจัดจังหวะยังคงทำงานอยู่ ใน Power-Down mode ค่าของ RAM และ SFRs' จะถูกเก็บเอาไว้ และจะหยุดการทำงานทั้งหมด ใน Power-Down mode สิ่งสำคัญที่สุด คือสามารถรับสัญญาณอินเทอร์รัปต์จากภายนอกได้ ทำให้ผู้ใช้สามารถลดการใช้พลังงานได้มากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.3 ส่วนประกอบของ MPC82G516A

- 1) โครงสร้างโดยทั่วไปที่สอดคล้องกับไมโครคอนโทรลเลอร์เบอร์ 8051 มีดังนี้
  - สามารถทำงานกับชุดคำสั่งของ 8051 ได้
  - มีหน่วยความจำภายใน(Internal Scratch-pad RAM) ขนาด 256 ไบต์
  - พื้นที่หน่วยความจำภายนอก 64 กิโลไบต์
  - พอร์ต I/O ขนาด 8 บิตรับส่งข้อมูลได้สองทิศทาง 4 พอร์ต
  - ตัวจับเวลา/ตัวนับ(Timer/Counters) ขนาด 16 บิต จำนวน 3 ตัว
  - มีพอร์ตอนุกรมที่สามารถรับส่งข้อมูลแบบพลูดูเพล็กซ์
  - สามารถอินเทอร์รัปต์ได้ 14 แหล่ง
  - ทำงานในโหมดประหยัดพลังงาน : Idle Mode & Power-Down Mode
- 2) หน่วยความจำภายในแบบแฟลช(Flash) 64 กิโลไบต์ สำหรับเก็บโปรแกรม
- 3) หน่วยความจำแรมเพิ่มเติม(XRAM) บนชิปขนาด 1024 ไบต์
- 4) พอร์ต I/O ที่สามารถใช้เป็น bit-address ติดต่อกับหน่วยความจำภายนอก(P4)
- 5) การใช้งานพอร์ต I/O สามารถใช้ได้หลายแบบ
  - พอร์ตพิเศษ 2 ทิศทาง(Quasi-bidirectional Output)
  - พอร์ตที่รับกระแสไฟฟ้าจากภายนอกเข้าไปในตัวได้(Open-drain Output)
  - พอร์ตอินพุตสำหรับเท่านั้น(Input-only)
  - พอร์ตสำหรับขับกระแสสูงๆ(Push-pull Output)
- 6) เพิ่มสัญญาณการขัดจังหวะภายนอกอีกสองแหล่งคือ /INT2 และ /INT3
- 7) เพิ่มความสามารถในการนับถอยหลังใน Timer2
- 8) โปรแกรมวงจรพิเศษ(Programmable Counter Array : PCA)
  - Capture Mode
  - Software Timer Mode 16 bit
  - High Speed Output Mode
  - PWM (Pulse Width Modulator) Mode
- 9) มีพอร์ตสื่อสารแบบ UART แบบที่สอง
- 10) มีการเชื่อมต่อแบบ SPI Interface (Master/Slave Mode)

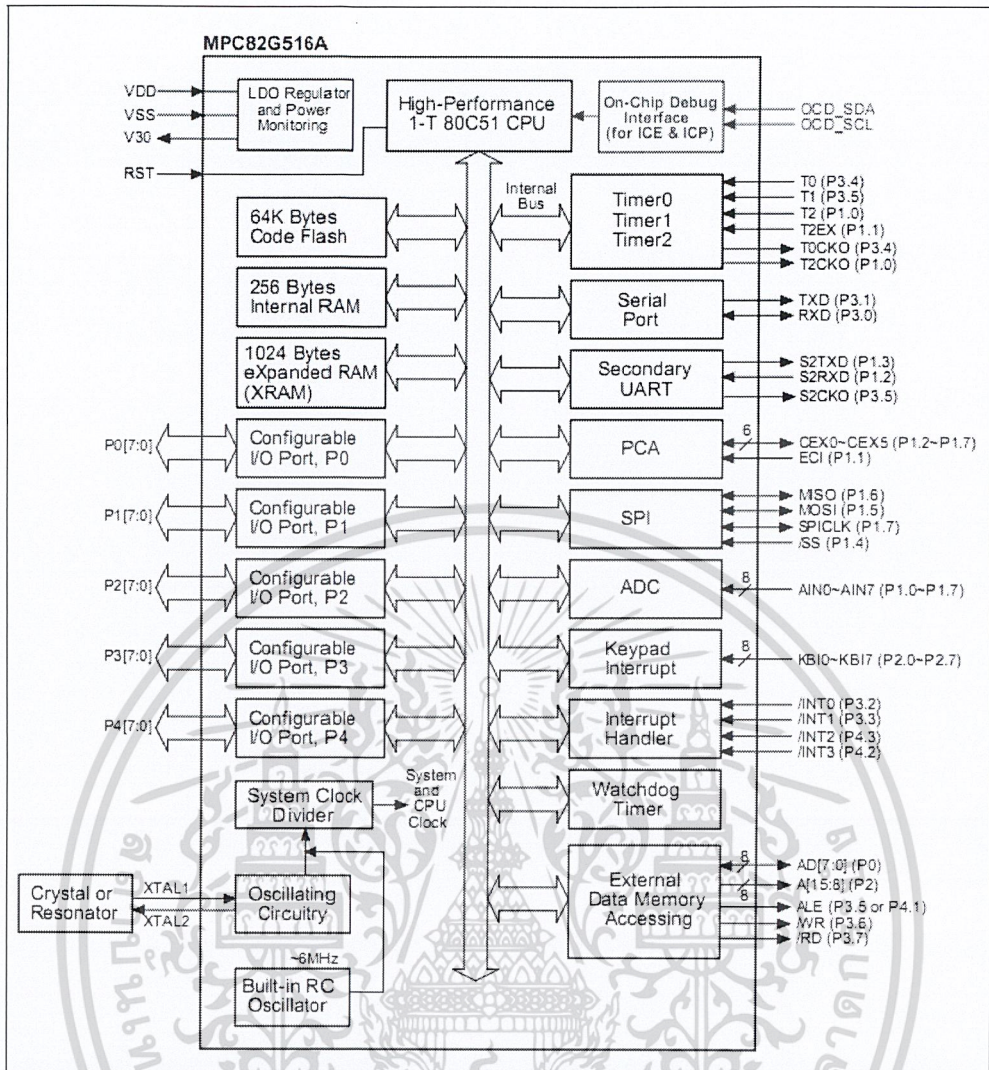
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 11) ตัวแปลงสัญญาณอนาล็อกเป็นดิจิทัล(ADC) 10 บิต สามารถมัลติเพล็กซ์ได้ 8 ช่องสัญญาณ
- 12) รับอินเทอร์รัปต์จากคีย์บอร์ดได้ 8 อินพุต
- 13) สามารถรับสัญญาณอินเทอร์รัปต์จากภายนอก ขณะที่พลังงานลดต่ำลง
- 14) สามารถโปรแกรมการสร้างสัญญาณนาฬิกาเพื่อส่งออกทาง output ได้
- 15) มีวอตช์ดอกไทมเมอร์(Watchdog Timer)
- 16) รีจิสเตอร์แบบ 2 ทาง ใช้อ้างตำแหน่งหน่วยความจำ(Dual DPTR)
- 17) การติดต่อกับหน่วยความจำภายนอกจะใช้คำสั่ง 'MOVX'
- 18) สามารถกำหนดระบบสัญญาณนาฬิกาเพื่อลดการใช้พลังงาน
- 19) ใช้การโปรแกรมแบบ ISP (In-System Programming) & ICP (In-Circuit Programming) ไปอัปเดตโปรแกรมในหน่วยความจำ
- 20) สามารถโปรแกรมแบบ IAP (In-Application Programming) ได้
- 21) สามารถอ่านเขียนแฟลชได้ 20,000 ครั้ง
- 22) สามารถใช้ความถี่การประมวลผลได้สูงถึง 24 เมกะเฮิร์ต
- 23) ใช้แหล่งจ่ายไฟ 2.4V ~ 3.6V (สำหรับระบบ 3.3V) หรือ 2.7V ~ 5.5V (สำหรับระบบ 5V)
- 24) ช่วงของอุณหภูมิ : -40 ถึง +85 องศาเซลเซียส
  - แพ็คเกจ : PDIP40, PLCC44, PQFP44, LQFP48 และ SSOP28

## 2.4 การทำงานของ MPC82G516A

ในรูปที่ 2.1 เป็นโครงสร้างภายในแสดงบล็อกไดอะแกรมแสดงการทำงานของ MPC82G516A โดยจะสรุปส่วนสำคัญของอุปกรณ์ ซึ่งผู้ใช้สามารถค้นหาการทำงานร่วมกับอุปกรณ์ต่อพ่วงจากไดอะแกรมได้ง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

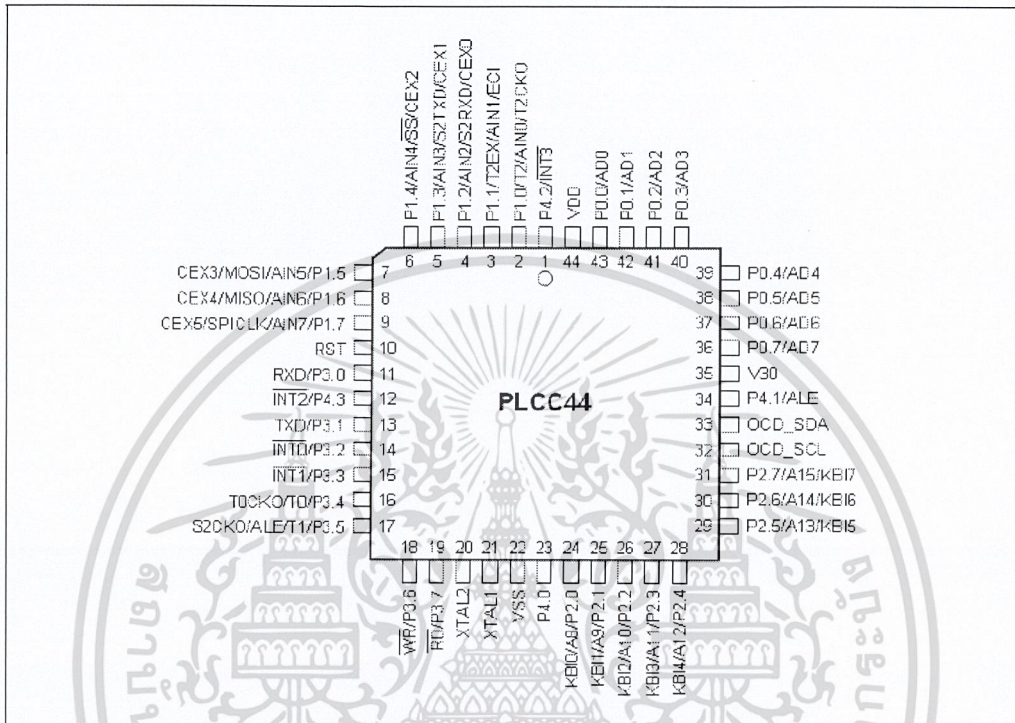


รูปที่ 2.1 บล็อกไดอะแกรมแสดงการทำงานของ MPC82G516A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.5 รูปแบบตัวถังและการขา

ไมโครคอนโทรลเลอร์เบอร์ MPC82G516A มีตัวถังการจัดขาหลายรูปแบบ เช่น PDIP 40 ขา, SSOP 28 ขา และ PLCC 44 ขา ซึ่งในที่นี้ใช้แบบ PLCC 44 ขา ซึ่งแสดงการทำงานได้ดังนี้



รูปที่ 2.2 แสดงการทำงานของ PLCC 44 ขา

รายละเอียดของขาต่างๆ

หน้าที่การทำงานของขาต่างๆ สรุปได้ดังตารางที่ 2.1

ตารางที่ 2.1 รายละเอียดของขาต่างๆ

MNEMONIC	40-Pin PDIP	28-Pin SSOP	44-Pin PLCC	I/O Type	รายละเอียด
P0.0 (Alt. Fun.) AD0	39	-	37	I/O I/O	-ใช้เป็นขาพอร์ต P0 bit-0 -แปลงสัญญาณนอกเป็น ดิจิทัล AD0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MNEMONIC	40-Pin PDIP	28-Pin SSOP	44-Pin PLCC	I/O Type	รายละเอียด
P0.1 (Alt. Fun.) AD1	38	27	36	I/O I/O	-ใช้เป็นขาพอร์ต P0 bit-1 -แปลงสัญญาณอนาลอกเป็น ดิจิตอล AD1
P0.2 (Alt. Fun) AD2	37	26	35	I/O I/O	-ใช้เป็นขาพอร์ต P0 bit-2 -แปลงสัญญาณอนาลอกเป็น ดิจิตอล AD2
P0.3 (Alt. Fun) AD3	36	25	34	I/O I/O	-ใช้เป็นขาพอร์ต P0 bit-3 -แปลงสัญญาณอนาลอกเป็น ดิจิตอล AD3
P0.4 (Alt. Fun) AD4	35	24	33	I/O I/O	-ใช้เป็นขาพอร์ต P0 bit-4 -แปลงสัญญาณอนาลอกเป็น ดิจิตอล AD4
P0.5 (Alt. Fun) AD5	34	-	32	I/O I/O	-ใช้เป็นขาพอร์ต P0 bit-5 -แปลงสัญญาณอนาลอกเป็น ดิจิตอล AD5
P0.6 (Alt. Fun) AD6	33	23	31	I/O I/O	-ใช้เป็นขาพอร์ต P0 bit-6 -แปลงสัญญาณอนาลอกเป็น ดิจิตอล AD6
P0.7 (Alt. Fun) AD7	32	-	30	I/O I/O	-ใช้เป็นขาพอร์ต P0 bit-7 -แปลงสัญญาณอนาลอกเป็น ดิจิตอล AD7
P1.0 (Alt. Fun) T2 (Alt. Fun) AIN0 (Alt. Fun) T2CKO	1	2	40	I/O I I O	-ใช้เป็นขาพอร์ต P1 bit-0 -ตัวจับเวลา ตัวนับตัวที่ 2 T2 -แปลงสัญญาณอนาลอกเป็น ดิจิตอล -เป็นสัญญาณนาฬิกาโปรแกรม ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MNEMONIC	40-Pin PDIP	28-Pin SSOP	44-Pin PLCC	I/O Type	รายละเอียด
P1.1 (Alt. Fun) T2EX (Alt. Fun) AIN1 (Alt. Fun) ECI	2	-	41	I/O I I I	-ใช้เป็นขาพอร์ต P1 bit-1 -ตัวจับเวลา ตัวนับตัวที่ 2 -แปลงสัญญาณอนาลอกเป็น ดิจิตอล
P1.2 (Alt. Fun) AIN2 (Alt. Fun) S2RXD (Alt. Fun) CEX0	3	3	42	I/O I I I/O	-ใช้เป็นขาพอร์ต P1 bit-2 -แปลงสัญญาณอนาลอกเป็น ดิจิตอล -รับข้อมูลทางพอร์ตอนุกรม
P1.3 (Alt. Fun) AIN3 (Alt. Fun) S2TXD (Alt. Fun) CEX1	4	4	43	I/O I O I/O	-ใช้เป็นขาพอร์ต P1 bit-3 -แปลงสัญญาณอนาลอกเป็น ดิจิตอล -ส่งข้อมูลทางพอร์ตอนุกรม
P1.4 (Alt. Fun) AIN4 (Alt. Fun) /SS (Alt. Fun) CEX2	5	-	44	I/O I I I/O	-ใช้เป็นขาพอร์ต P1 bit-4 -แปลงสัญญาณอนาลอกเป็น ดิจิตอล -รับ-ส่งข้อมูล
P1.5 (Alt. Fun) AIN5 (Alt. Fun) MOSI (Alt. Fun) CEX3	6	5	1	I/O I I/O I/O	-ใช้เป็นขาพอร์ต P1 bit-5 -แปลงสัญญาณอนาลอกเป็น ดิจิตอล
P1.6 (Alt. Fun) AIN6 (Alt. Fun) MISO (Alt. Fun) CEX4	7	-	2	I/O I I/O I/O	-ใช้เป็นขาพอร์ต P1 bit-6 -แปลงสัญญาณอนาลอกเป็น ดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MNEMONIC	40-Pin PDIP	28-Pin SSOP	44-Pin PLCC	I/O Type	รายละเอียด
P1.7 (Alt. Fun) AIN7 (Alt. Fun) SPICLK (Alt. Fun) CEX5	8	-	3	I/O I I/O I/O	-ใช้เป็นขาพอร์ต P1 bit-7 -แปลงสัญญาณอนาลอกเป็น ดิจิตอล
P2.0 (Alt. Fun) A8 (Alt. Fun) KBI0	21	15	18	I/O O I	-ใช้เป็นขาพอร์ต P2 bit-0 -เชื่อมต่อกับ address -รับข้อมูลเข้าจาก keypad 0
P2.1 (Alt. Fun) A9 (Alt. Fun) KBI1	22	16	19	I/O O I	-ใช้เป็นขาพอร์ต P2 bit-1 -เชื่อมต่อกับ address -รับข้อมูลเข้าจาก keypad 1
P2.2 (Alt. Fun) A10 (Alt. Fun) KBI2	23	-	20	I/O O I	-ใช้เป็นขาพอร์ต P2 bit-2 -เชื่อมต่อกับ address -รับข้อมูลเข้าจาก keypad 2
P2.3 (Alt. Fun) A11 (Alt. Fun) KBI3	24	17	21	I/O O I	-ใช้เป็นขาพอร์ต P2 bit-3 -เชื่อมต่อกับ address -รับข้อมูลเข้าจาก keypad 3
P2.4 (Alt. Fun) A12 (Alt. Fun) KBI4	25	18	22	I/O O I	-ใช้เป็นขาพอร์ต P2 bit-4 -เชื่อมต่อกับ address -รับข้อมูลเข้าจาก keypad 4
P2.5 (Alt. Fun) A13 (Alt. Fun) KBI5	26	19	23	I/O O I	-ใช้เป็นขาพอร์ต P2 bit-5 -เชื่อมต่อกับ address -รับข้อมูลเข้าจาก keypad 5
P2.6 (Alt. Fun) A14 (Alt. Fun) KBI6	27	-	24	I/O O I	-ใช้เป็นขาพอร์ต P2 bit-6 -เชื่อมต่อกับ address -รับข้อมูลเข้าจาก keypad 6
P2.7 (Alt. Fun) A15 (Alt. Fun) KBI7	28	-	25	I/O O I	-ใช้เป็นขาพอร์ต P2 bit-7 -เชื่อมต่อกับ address -รับข้อมูลเข้าจาก keypad 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MNEMONIC	40-Pin PDIP	28-Pin SSOP	44-Pin PLCC	I/O Type	รายละเอียด
P3.0 (Alt. Fun) RXD	10	7	5	I/O I/O	-ใช้เป็นขาพอร์ต P3 bit-0 -รับข้อมูลทางพอร์ตอนุกรม
P3.1 (Alt. Fun) TXD	11	8	7	I/O O	-ใช้เป็นขาพอร์ต P3 bit-1 -ส่งข้อมูลทางพอร์ตอนุกรม
P3.2 (Alt. Fun) /INT0	12	9	8	I/O I	-ใช้เป็นขาพอร์ต P3 bit-2 - interrupt ภายนอกหมายเลข 0
P3.3 (Alt. Fun) /INT1	13	10	9	I/O I	-ใช้เป็นขาพอร์ต P3 bit-3 - interrupt ภายนอกหมายเลข 1
P3.4 (Alt. Fun) T0 (Alt. Fun) T0CKO	14	-	10	I/O I O	-ใช้เป็นขาพอร์ต P3 bit-4 -ตัวจับเวลา ตัวนับตัวที่ 0 -เป็นสัญญาณนาฬิกา
P3.5 (Alt. Fun) T1 (Alt. Fun) ALE (Alt. Fun) S2CKO	15	11	11	I/O I O O	-ใช้เป็นขาพอร์ต P3 bit-5 -ตัวจับเวลา ตัวนับตัวที่ 1 -เป็นสัญญาณที่ติดต่อกับ หน่วยความจำภายนอก -เป็นสัญญาณนาฬิกา
P3.6 (Alt. Fun) /WR	16	-	12	I/O O	-สัญญาณเขียนข้อมูล หน่วยความจำภายนอก
P3.7 (Alt. Fun) /RD	17	-	13	I/O O	-สัญญาณอ่านข้อมูล หน่วยความจำภายนอก
P4.0	-	-	17	I/O	-ใช้เป็นขาพอร์ต P4 bit-0
P4.1 (Alt. Fun) ALE	-	-	28	I/O O	-ใช้เป็นขาพอร์ต P4 bit-1 -เป็นสัญญาณที่ติดต่อกับ หน่วยความจำภายนอก
P4.2 (Alt. Fun) /INT3	-	1	39	I/O I	-ใช้เป็นขาพอร์ต P4 bit-2 - interrupt ภายนอกหมายเลข 3
P4.3 (Alt. Fun) /INT2	-	-	6	I/O I	-ใช้เป็นขาพอร์ต P4 bit-3 - interrupt ภายนอกหมายเลข 2

MNEMONIC	40-Pin PDIP	28-Pin SSOP	44-Pin PLCC	I/O Type	รายละเอียด
P4.4	-	-	-	I/O	-ใช้เป็นขาพอร์ต P4 bit-4
P4.5	-	-	-	I/O	-ใช้เป็นขาพอร์ต P4 bit-5
P4.6	-	-	-	I/O	-ใช้เป็นขาพอร์ต P4 bit-6
P.4.7	-	-	-	I/O	-ใช้เป็นขาพอร์ต P4 bit-7
OCD_SDA	30	21	27	I/O	-เป็นขารับส่งข้อมูลแบบอนุกรม
OCD_SCL	29	20	26	I	-เป็นขาอินพุตสัญญาณนาฬิกา อนุกรม
XTAL1	19	13	15	I	-ใช้ต่อกับคริสตอล สำหรับ กำหนดสัญญาณนาฬิกาให้ ไมโครคอนโทรลเลอร์
XTAL2	18	12	14	O	-ใช้ต่อกับคริสตอล สำหรับ กำหนดสัญญาณนาฬิกาให้ ไมโครคอนโทรลเลอร์
RST	9	6	4	I	-สำหรับรีเซ็ตระบบ
V30	31	22	29	O	-เป็นสัญญาณเอาต์พุตจากวงจร กำเนิด LDO ภายใน
VDD	40	28	38	I	-แหล่งจ่ายไฟ
VSS	20	14	16	I	-ขาต่อกราวด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.6 การสร้างโค้ด

ในการสร้างโค้ดจากส่วนไคอะแกรมที่แสดงโพลีชาร์ต อาศัยหลักการดังนี้

### พื้นฐานการสร้างโค้ด

เมื่อผู้ใช้สามารถวางผังงานที่ต้องการลงไปไคอะแกรม เมื่อวาดเสร็จแล้ว จะทำการตรวจสอบความถูกต้องของโพลีชาร์ตที่ผู้ใช้วาด จากนั้นเมื่อโพลีชาร์ตถูกต้องแล้ว ทำการสร้างโค้ดเป็นภาษา C สำหรับไมโครคอนโทรลเลอร์ตระกูล MCS-51 ขั้นตอนในการสร้างเป็นโค้ดภาษา C สำหรับไมโครคอนโทรลเลอร์ตระกูล MCS-51 นี้ ได้ใช้ทฤษฎีทางการตรวจสอบรูปแบบเรียกว่า LR Parsing ต่อไปขอกล่าวถึงพื้นฐานและหลักการของการตรวจสอบรูปแบบดังกล่าว การ Parse เป็นการทดสอบว่าโปรแกรมที่ Input เข้าามี Syntax ที่ถูกต้องหรือไม่ ซึ่งมีวิธีการ Parse ได้หลายวิธีในที่นี้ เราจะใช้ LR Parsing แต่ก่อนจะอธิบาย LR Parsing จะกล่าวถึง Shift-Reduce Parsing ก่อน

### Shift-Reduce Parsing

ในการอิมพลีเมนต์ จะใช้ สตЕКเพื่อเก็บ Symbol ที่อ่านเข้ามา และใช้อินพุตบัฟเฟอร์ เพื่อเก็บอินพุต ที่จะทำการ Parse เราจะใช้เครื่องหมาย \$ บอกลำดับตำแหน่งล่างสุดของสตЕК และบอกตำแหน่งขวาสุดของอินพุต

ตัวอย่างเช่น เมื่อเริ่มต้นที่สตЕКว่างเปล่า และสตริง  $w$  เป็นอินพุต จะเป็นตามนี้

Stack	Input
\$	$w$ \$

Parser จะทำการ Shift จนกระทั่ง Stack บรรจุ Symbol ที่ใช้เริ่มต้น  $w$  และ Reduce  $w$  ใน Stack จนไม่มีเหลืออยู่ใน Stack ดังเช่น ตัวอย่างข้างล่าง Reduce  $w$  เป็น  $s$  ซึ่ง  $s$  เป็น Start Symbol

Stack	Input
\$ $s$	$s$

จากนั้นแล้ว Parser จะถือว่าเสร็จสิ้นการทำ Parsing

ตัวอย่างเช่น ในการทำ Parsing Input string :  $id1 + id2 * id3$

ตามหลัก Grammar

- (1)  $E \rightarrow E + E$
- (2)  $E \rightarrow E * E$
- (3)  $E \rightarrow (E)$
- (4)  $E \rightarrow id$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 การ Parse แบบ Shift-Reduce

STACK	INPUT	ACTION
(1) \$	id1 + id2 * id3\$	Shift
(2) \$id1	+ id2 * id3\$	Reduce by E -> id
(3) \$E	+ id2 * id3\$	Shift
(4) \$E +	id2 * id3\$	Shift
(5) \$E + id2	* id3\$	Reduce by E -> id
(6) \$E + E	* id3\$	Shift
(7) \$E + E *	id3\$	Shift
(8) \$E + E * id3	\$	Reduce by E -> id
(9) \$E + E * E	\$	Reduce by E -> E * E
(10) \$E + E	\$	Reduce by E -> E + E
(11) \$E	\$	Accept

ถึงแม้ว่าการกระทำหลักของ Parser จะเป็นการ Shift และ Reduce แต่โดยแท้จริงแล้วมี 4 Action ที่เกิดขึ้นได้ ดังนี้

- Shift เป็นการทำให้ Input Symbol ถัดไปถูกหยิบใส่ลงบน Stack
- Reduce ทำให้ Parser รู้ว่าตำแหน่งบนสุดของ Stack นั้นมีค่าตรงกับ ด้านขวาของ Production Rule ใดๆ ของ Grammar ที่ใช้หรือไม่ แล้วทำการเปลี่ยนค่าบนสุดของ Stack ให้ตรงตามกับ Grammar
- Accept เมื่อ Parser ทำเสร็จสิ้น Stack บรรจุ Start Symbol
- Error เมื่อ Parser พบ Syntax ที่ผิดพลาดแล้วเรียก Routine ของการแก้ไขข้อผิดพลาด นั้น

ตัวอย่าง Grammar ของ expression ด้วย operator + และ \* หรืออีกชื่อหนึ่งที่เรียกว่า production :

- (1)  $E \rightarrow E + T$
- (2)  $E \rightarrow T$
- (3)  $T \rightarrow T * F$
- (4)  $T \rightarrow F$
- (5)  $F \rightarrow (E)$

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 Parsing ของ Grammar expression

STATE	action						goto		
	id	+	*	(	)	\$	E	T	F
0	s5			s4			1	2	3
1		s6				acc			
2		r2	s7		r2	r2			
3		r4	r4		r4	r4			
4	s5			s4			8	2	3
5		r6	r6		r6	r6			
6	s5			s4				9	3
7	s5			s4					10
8		s6			s11				
9		r1	s7		r1	r1			
10		r3	r3		r3	r3			
11		r5	r5		r5	r5			

- si หมายถึง การ Shift และ State ของ Stack มีค่าเท่ากับ i
- rj หมายถึง การ Reduce โดย Production ตาม Grammar หมายเลข j
- Acc หมายถึง Accept
- Blank หมายถึง Error

จากตัวอย่างนี้ Input ที่เป็น  $id * id + id$  มีลำดับแต่ละชั้นของ Stack และแต่ละชั้นของ Input ที่แสดงไว้ดังตารางข้างล่างนี้ ตัวอย่าง เช่น ในบรรทัดแรก LR Parser อยู่ในบรรทัดแรก ด้วย ค่า id เป็นค่า Input ค่าแรก และ Action ในตาราง Expression Grammar State ที่ 0 และคอลัมภ์ id เป็น s5 หมายถึงให้ทำการ Shift ลงบน Stack ด้วย state 5 ซึ่งเกิดขึ้นให้เห็นในบรรทัดที่ (2) Token แรกเป็น id และ state 5 แล้วมีผลให้ \* กลายเป็น input ปัจจุบันแทนและ Action ในตาราง Expression Grammar state ที่ 5 คอลัมภ์ \* เป็น r6 หมายถึง ให้ทำการ Reduce โดยใช้ Grammar ที่ (6)  $F \rightarrow id$  มีผลให้ 2 Symbol ถูก Pop ออกจาก Stack คือค่า id กับค่า 5 ถูก Pop ออกจาก Stack และแทนที่ใน Stack ด้วย F และ 3 ที่ได้ค่ามาจากตาราง Expression Grammar ซึ่งดูว่าเมื่อ Top Stack มีค่าเป็น 0 Symbol F ที่ได้จะ Shift state 3 (ในช่อง goto F แถว 0 เข้าไปใน Stack) ดังรูปแบบบรรทัดที่ (3) และอันที่เหลือมีการทำงานคล้าย ๆ กันดังที่ทำมาข้างต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 การ Move ของ LR Parser บน  $id * id + id$

STACK	INPUT	ACTION
(1) 0	id * id + id \$	Shift
(2) 0 id 5	* id + id \$	Reduce by F -> id
(3) 0 F 3	* id + id \$	Reduce by T -> F
(4) 0 T 2	* id + id \$	Shift
(5) 0 T 2 * 7	id + id \$	Shift
(6) 0 T 2 * 7 id 5	+ id \$	Reduce by F -> id
(7) 0 T 2 * 7 F 10	+ id \$	Reduce by T -> T * F
(8) 0 T 2	+ id \$	Reduce by E -> T
(9) 0 E 1	+ id \$	Shift
(10) 0 E 1 + 6	id \$	Shift
(11) 0 E 1 + 6 id 5	\$	Reduce by F -> id
(12) 0 E 1 + 6 F 3	\$	Reduce by T -> F
(13) 0 E 1 + 6 T 9	\$	Reduce by E -> E + T
(14) 0 E 1	\$	Accept

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

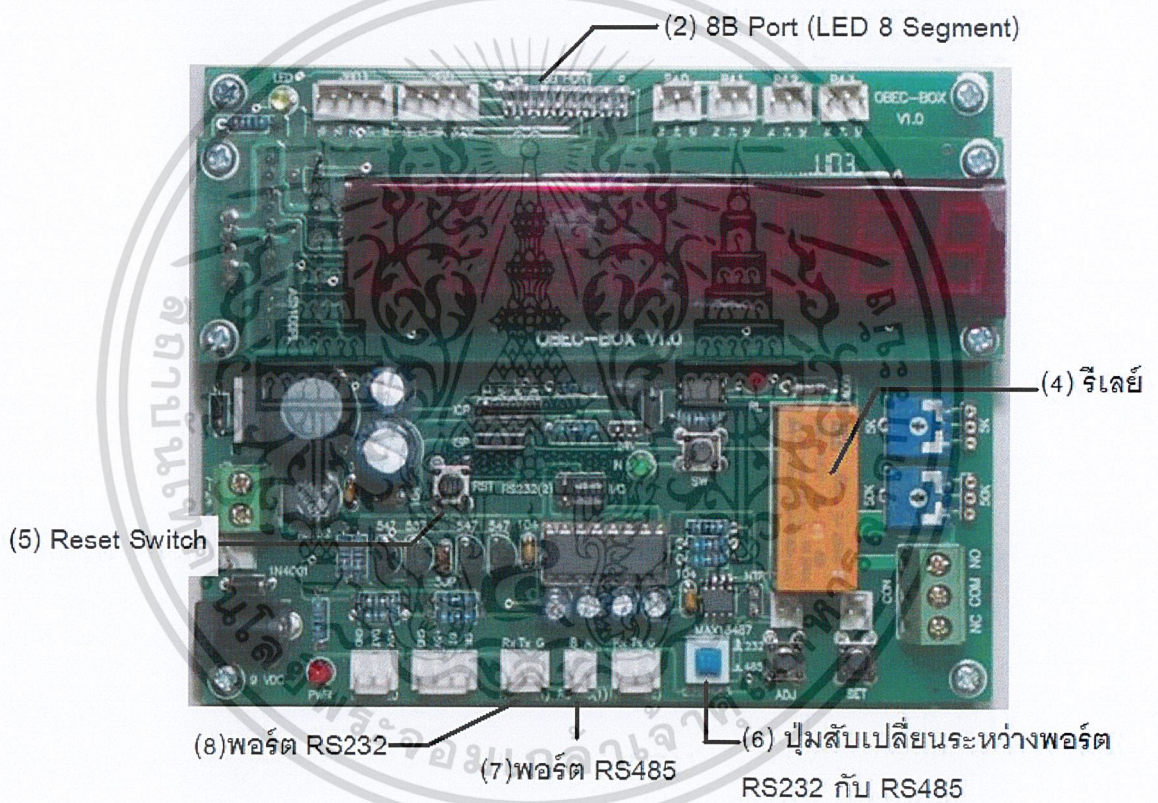
## บทที่ 3

### ขั้นตอนและการออกแบบระบบ

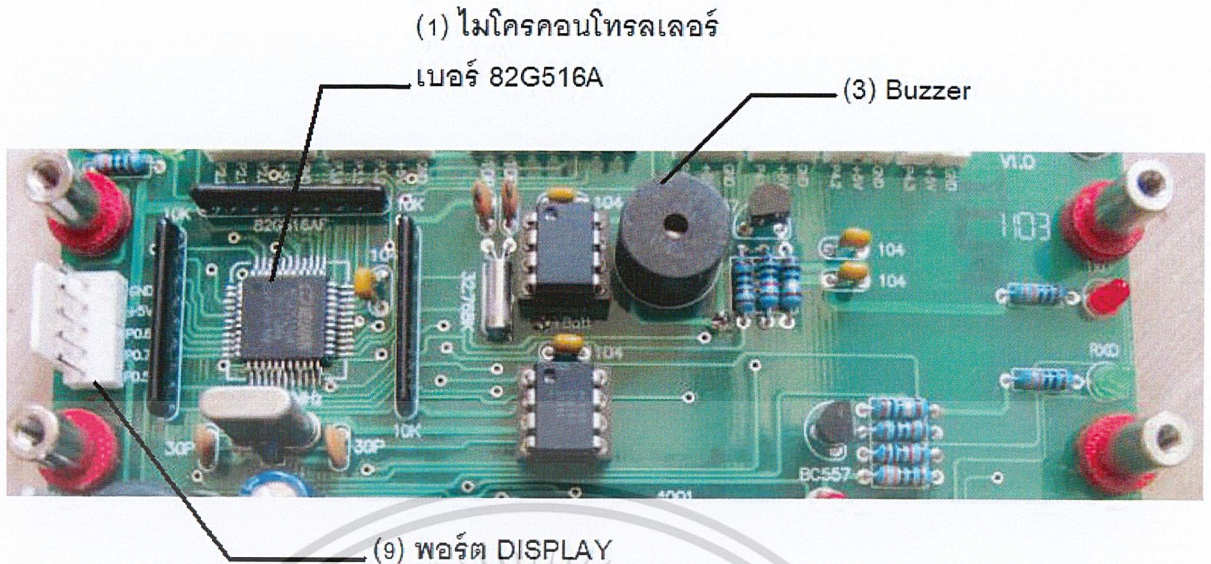
#### 3.1 ขั้นตอนการดำเนินงาน

##### 3.1.1 ศึกษารายละเอียดเกี่ยวกับ MPC82G516A

ซึ่งเป็น ไมโครคอนโทรลเลอร์ชิปเดี่ยว รวมทั้งศึกษาส่วนประกอบต่างๆของบอร์ด ทดลองถึงการทำงานและฟังก์ชันในการเรียกใช้ในแต่ละส่วน เช่น BUZZER เป็นฟังก์ชันเกี่ยวกับ เสียง และ DISNUM เป็นฟังก์ชันแสดงตัวเลข เป็นต้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 ส่วนประกอบต่างๆบนบอร์ดทดลอง

บอร์ดทดลองจากรูป 3.1 นี้มีส่วนประกอบหลายส่วนด้วยกัน ในที่นี้จะกล่าวถึงส่วนประกอบสำคัญที่ใช้ในตัวอย่างการทดลองของปริญญาโทฉบับนี้เท่านั้น ได้แก่

- 1) ไมโครคอนโทรลเลอร์เบอร์ 82G516A ซึ่งประกอบด้วยตัวถังและขา 44 ขา
- 2) 8B Port ใช้ในการต่อหลอดไฟ LED 8 เซ็กเมนต์
- 3) Buzzer คือลำโพงแบบแม่เหล็กหรือ แบบเปียโซที่มีวงจรถูกกำเนิดความถี่ (oscillator) อยู่ในตัวเมื่อป้อนแรงดันสามารถกำเนิดเสียงได้ด้วยตัวเอง แต่ไม่สามารถเปลี่ยนความถี่ของเสียงได้
- 4) รีเลย์ คือ อุปกรณ์อิเล็กทรอนิกส์ที่ทำหน้าที่ ตัด-ต่อวงจร คล้ายกับสวิตช์ โดนมียุทธการงานคล้ายกับขดลวดแม่เหล็กไฟฟ้า ในที่นี้เราจะใช้รีเลย์ในการเปิดปิดไฟ
- 5) Reset Switch
- 6) ปุ่มสับเปลี่ยนระหว่างพอร์ต RS232 กับ RS485
- 7) พอร์ต RS485
- 8) พอร์ต RS232 พอร์ตนี้ใช้ในการเชื่อมต่อระหว่างคอมพิวเตอร์กับบอร์ดทดลอง
- 9) พอร์ต DISPLAY เป็นพอร์ตที่ใช้เชื่อมต่อกับหน้าจอแสดงผลตัวเลขจำนวน 8 หลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.2 ศึกษาการเขียนโปรแกรมจาวาเชิงวัตถุ

เนื่องจากโปรแกรมโพลีโค้ดเป็นโปรแกรมที่มีขนาดใหญ่และมีความซับซ้อนในแง่ของโปรแกรม จึงควรมีการเขียนโปรแกรมในรูปแบบ Object Oriented Programming ซึ่งแนวคิดนี้ช่วยแก้ปัญหาในเรื่องการนำกลับมาใช้ใหม่ การแก้ไข และการขยายเพื่อเพิ่มประสิทธิภาพให้กับโปรแกรมอีกด้วย

### 3.1.3 ออกแบบหน้าจอส่วนติดต่อกับผู้ใช้และพัฒนาไอคอนงานเฉพาะ

สำหรับการออกแบบโปรแกรมพื้นฐานเพื่อรองรับไมโครคอนโทรลเลอร์โดยมีการแบ่งหน้าจอออกเป็น 5 ส่วน ได้แก่

#### 1) Menu Bar ประกอบด้วย

- File ประกอบด้วยเมนูย่อย ได้แก่ New Project, Open Diagram, Save Code, Save Diagram, Close Project และ Exit
- View ประกอบด้วยเมนูย่อย ได้แก่ Toolbars, Documentation, Status Bar ซึ่งเป็นตัวเลือกแบบ Check box
- Setup ประกอบด้วยเมนูย่อย ได้แก่ Series Board
- Command ประกอบด้วยเมนูย่อย ได้แก่ Start, Stop, Assignment, Flow, Input, Output, Condition, EndCondition, WhileLoop, EndWhileLoop, ForLoop, EndForLoop, DoWhileLoop, EndDoWhileLoop, Buzzer, LED, Delay, number และ Relay
- Tools ประกอบด้วยเมนูย่อย ได้แก่ CheckModel, CheckSyntax และ GenerateCode

#### 2) Tool Bar ประกอบด้วย

- New Project
- Check Model
- Check Syntax
- Generate Code

3) Model Bar ประกอบด้วยคำสั่งที่อยู่ใน Command ในส่วนของ Menu Bar เพื่อสะดวกในการใช้งานเพราะลดความซับซ้อนของฟังก์ชันเมาส์และช่วยในการจดจำของผู้ใช้

4) Diagram คือส่วนที่แสดงไดอะแกรม ในที่นี้คือโพลีชาร์ตที่ผู้ใช้สร้างขึ้นมานั่นเอง

5) Code Area คือส่วนที่แสดงโค้ดภาษา C โดยก่อนที่โปรแกรมจะแสดงส่วนนี้ได้

จะต้องทำการ Check Model, Check Syntax และ Generate Code ตามขั้นตอนก่อน หากมี

ข้อผิดพลาดจะมีข้อความแจ้งเตือนและโปรแกรมจะไม่สามารถแสดงโค้ดได้จนผู้ใช้จะทำการแก้ไข

จนถูกต้อง ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**3.1.4. ออกแบบและพัฒนาเมนู**

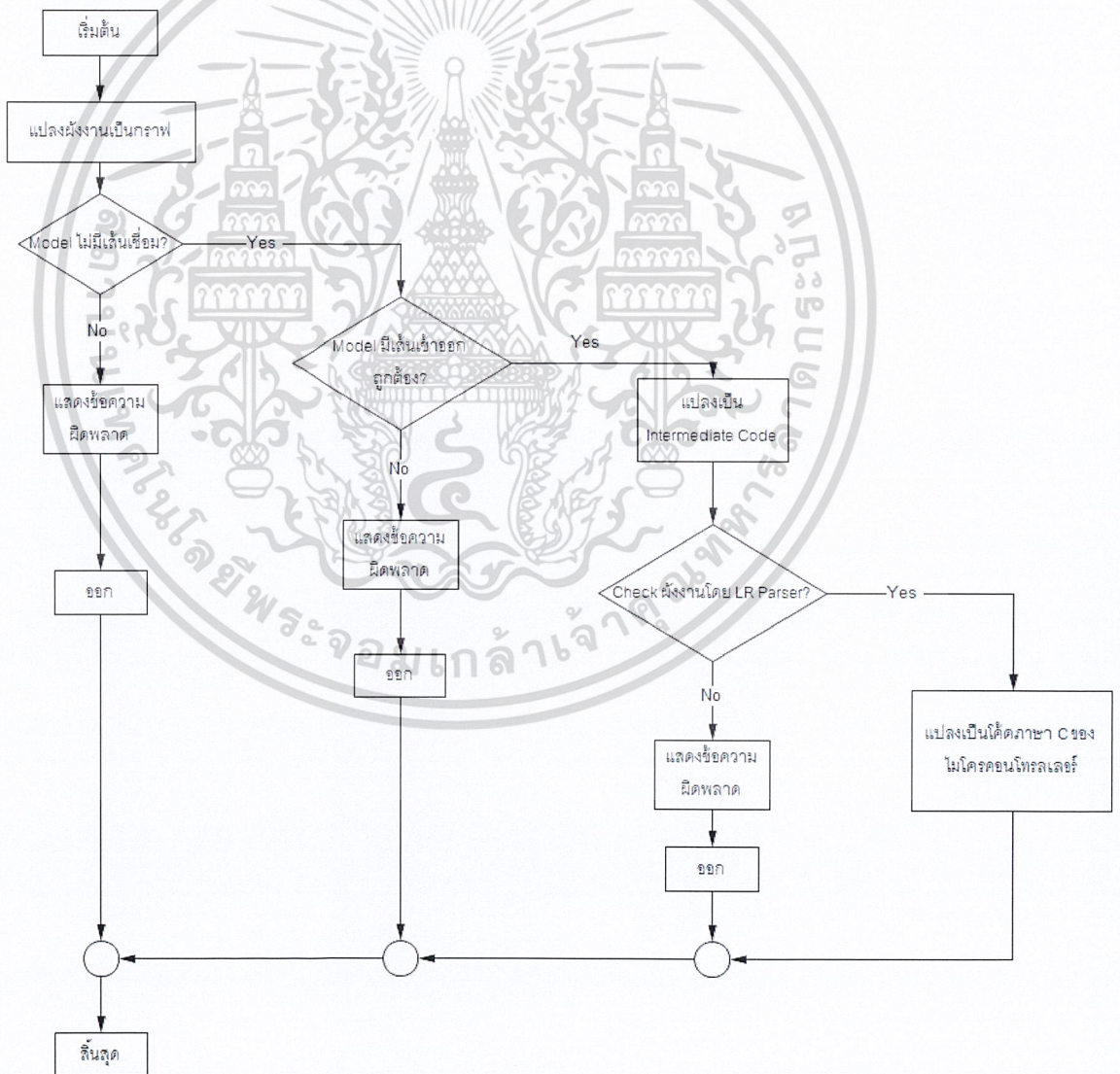
ช่วยสร้าง Configuration งานเฉพาะสำหรับไมโครคอนโทรลเลอร์ เช่น การทำงานในแต่ละพอร์ตและการติดต่อกับพอร์ตนั้นๆ เป็นต้น

**3.1.5. ปรับปรุง Property และสร้าง Dialog**

สำหรับองค์ประกอบในภาษา C ให้เหมาะกับการโปรแกรมไมโครคอนโทรลเลอร์

**3.1.6. สร้างกฎ Parser เพื่อตรวจสอบ Syntax ของการวาด**

เนื่องจากการเขียนโคแอดแกรม ผู้สามารถวาดองค์ประกอบลงในโคแอดแกรมได้โดยอิสระ ดังนั้นผู้ใช้เองอาจมีการวาดโคแอดแกรมที่ผิดรูปแบบได้ ดังนั้นก่อนการสร้างโค้ดภาษา C สำหรับไมโครคอนโทรลเลอร์ MPC82G516A จึงจำเป็นต้องทำการตรวจสอบว่าผู้ใช้ ได้เขียนโคแอดแกรมที่ถูกต้องตามรูปแบบหรือไม่ โดยขั้นตอนการตรวจสอบดังรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
**รูปที่ 3.2 อัลกอริทึมการตรวจสอบ Syntax ของผังงาน โดยรวม**  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1) การตรวจสอบ Model ของผังงาน

ในขั้นตอนนี้จะมีการตรวจสอบว่าในไดอะแกรมมีคอมโปเนนท์ ที่อยู่ลอย ๆ ตามลำพังโดยที่ไม่มีเส้นเชื่อมต่อเข้าออกเลขหรือไม่ ถ้ามีให้แสดงข้อความผิดพลาดออกมาให้ผู้ใช้ทำการกลับไปแก้ไขไดอะแกรมให้ถูกต้อง จากนั้น จะทำการตรวจสอบว่าคอมโปเนนท์ แต่ละชนิดว่ามีจำนวนเส้นเข้าและออกถูกต้องและครบตาม Syntax หรือไม่ โดยพิจารณา ดังนี้

- สัญลักษณ์จุดเริ่มต้น (Start Model) ต้องมีเส้นที่ออกจากตัวเองไม่มีเส้นเข้าตัวโมเดลเองและต้องมีเพียงเส้นเดียวเท่านั้น
- สัญลักษณ์จุดสิ้นสุด (Stop Model) ต้องมีแต่เส้นเข้าไม่มีเส้นออกจากตัวโมเดล
- สัญลักษณ์การป้อนข้อมูล (Input Model), สัญลักษณ์การแสดงผล (Output Model), สัญลักษณ์การป้อนข้อมูลของพอร์ต (Input Port), สัญลักษณ์การแสดงผลของพอร์ต (Output Port), สัญลักษณ์ของการหน่วงเวลา (Delay) จะต้องมีเส้นออกและเส้นเข้าหาโมเดลอย่างละ 1 เส้นเท่านั้น
- สัญลักษณ์การเลือก (Condition Model) ต้องมีเส้นออกจากตัว 2 เส้นเสมอแยกเป็น Yes /No และเข้าหาตัวโมเดล 1 เส้น
- สัญลักษณ์ปิดการเลือก (EndCondition Model) จะต้องมีเส้นเข้า 2 เส้นและออก 1 เส้น
- สัญลักษณ์การให้ค่า (Assignment Model), สัญลักษณ์การทำซ้ำ ทั้ง For Model, While Model, DoWhile Model ต้องมีเส้นออกและเส้นเข้าหาโมเดลอย่างละ 1 เส้นเท่านั้น
- สัญลักษณ์ปิดลูป EndWhile Model, EndFor Model, End DoWhile Model จะต้องมีเส้นเข้า 1 เส้นและออก 1 เส้นอยู่ด้วย

ถ้าจำนวนเส้นเข้าและออกไม่ครบดังกล่าว โปรแกรมจะแสดงข้อความผิดพลาดออกมาให้ผู้ใช้ทำการแก้ไขผังงานให้ถูกต้อง ขั้นตอนที่ไปเป็นการตรวจสอบดูว่าไดอะแกรมที่มีนั้นมีการเชื่อมต่อของคอมโปเนนท์ต่างๆ ที่ถูกต้องตามหลักของไวยากรณ์ภาษาC ของไมโครคอนโทรลเลอร์หรือไม่ ตัวอย่างเช่น ในการตรวจสอบผังงานในขั้นตอนนี้จะตรวจสอบว่า ConditionModel มี EndCondition Model หรือไม่ นั่นหมายถึงว่า เมื่อมี IF ก็ต้องมี ENDIF นั่นเอง เมื่อตรวจสอบแล้วพบว่าถูกต้องและครบแล้วจะถือว่าตรวจสอบโมเดล ในขั้นตอนนี้สิ้นสุด

## 2) การตรวจสอบ Syntax ของแต่ละโมเดล

ในขั้นแรกจะตรวจสอบดูว่าในแต่ละโมเดลนั้นค่าในตัวมันอยู่หรือไม่ หากโมเดลใดไม่มีค่าจะแสดงข้อความผิดพลาดให้ผู้ใช้ได้กลับไปกรอก Property ภายในตัวโมเดลนั้น ยกตัวอย่างเช่น หากอินพุตโมเดล ยังไม่มีการกำหนดทั้ง 3 อย่างต่อไปนี้คือ ชื่อของตัวแปร, ชนิดของตัวแปร, ค่าของตัวแปร จะมีข้อความแสดงบอกว่ายังไม่มีค่า ผู้ใช้จะต้องกรอกค่าต่างๆ เหล่านี้ให้ครบถ้วน

เสียก่อน เมื่อตรวจสอบว่าถูกต้องและครบถ้วนแล้วถือว่าการตรวจสอบ Syntax ในขั้นตอนนี้สิ้นสุดลง

### 3.1.7. สร้างโค้ดภาษา C

ขั้นตอนของการสร้างโค้ดภาษา C แบ่งเป็น 3 ขั้นตอน ได้แก่

#### 1) การตรวจสอบโมเดล

โดยจะมีการตรวจสอบว่าในไฟล์ชาร์ตมีโมเดล Start และโมเดล Stop อยู่หรือไม่ โดยมีการพิจารณาดังต่อไปนี้

1.1) จากการลากเส้นเชื่อมระหว่างโมเดลของไฟล์ชาร์ต จะมีการนำข้อมูลต่างๆ เก็บอยู่ในรูปของตาราง โดยที่ตารางนั้นจะเก็บข้อมูลของ ชื่อโมเดลต้นทาง, ชื่อของโมเดลปลายทาง, ตำแหน่งของโมเดลต้นทาง, ตำแหน่งของโมเดลปลายทาง, property ของเส้น, โค้ดของโมเดลต้นทาง, โค้ดของโมเดลปลายทาง

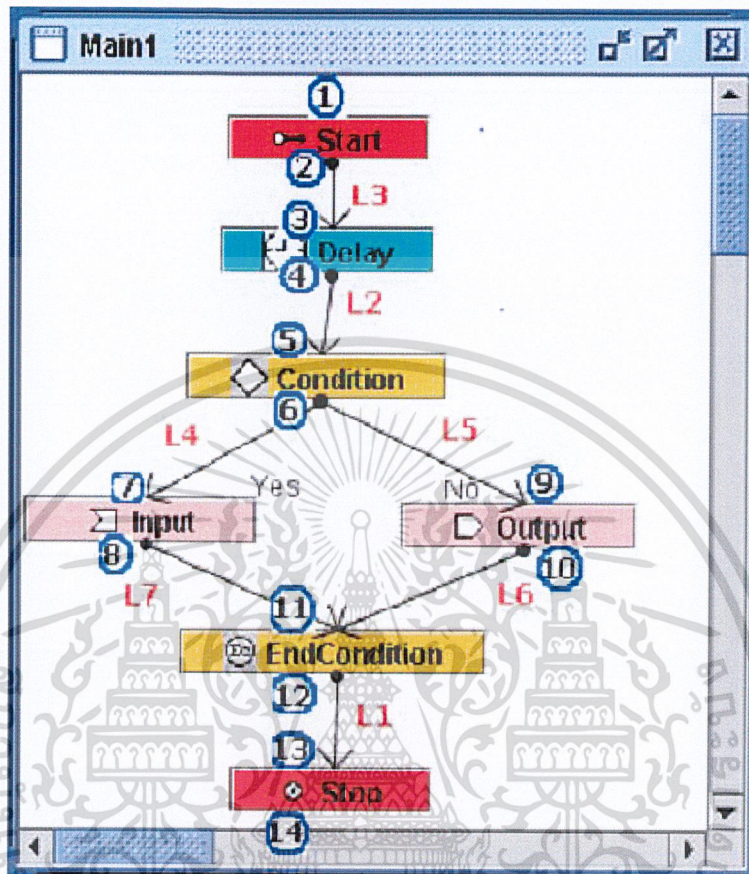
1.2) ทำการเรียงลำดับข้อมูลในตารางโดยให้เริ่มจากจุดเริ่มต้นคือ Start Model และไล่หาโมเดลต่อไปจากทิศทางของลูกศรและไปสิ้นสุดที่โมเดล Stop

1.3) ตรวจสอบว่าทุกโมเดลมีเส้นลากเข้า หรือเส้นลากออกครบและถูกต้องหรือไม่ โดยการหาจำนวนเส้นลากเข้า หรือลากออกหาได้จาก  
(จำนวนของโมเดลทั้งหมด - 1) + (จำนวนของโมเดล Condition \* 2)

1.4) ตรวจสอบว่าโมเดลทุกโมเดลวางถูกตำแหน่งหรือไม่ ในการตรวจสอบว่าโมเดลวางถูกตำแหน่งนั้น ใช้วิธีการของสแตก คือในขั้นตอนแรก จะต้องสร้างสแตกว่างๆ มารับโมเดลจากไฟล์ชาร์ต โมเดลที่สามารถเก็บลงในสแตกได้นั้นมีเพียง 4 โมเดลเท่านั้น ได้แก่ Condition, While, EndDoWhile และ For ส่วนการนำโมเดลออกจากสแตกนั้นจะเกิดขึ้นเมื่อโมเดล Condition คู่กับโมเดล EndCondition, โมเดล While คู่กับ EndWhile, โมเดล EndDoWhile คู่กับโมเดล DoWhile และ โมเดล For คู่กับ โมเดล EndFor

พิจารณาตัวอย่างที่ 3.1

ตัวอย่างที่ 3.1 ในตัวอย่างนี้มีการใช้โมเดล 7 โมเดล ได้แก่ Start, Condition, Input, EndCondition, Output, Delay และ Stop โดยมีการจำลองการวาดโฟลว์ชาร์ตดังรูป ที่ 3.3



รูปที่ 3.3 ตัวอย่างการวาดโฟลว์ชาร์ต

โดยจากโฟลว์ชาร์ตจะมีการเก็บข้อมูลในรูปแบบของตาราง 3.1 สังเกตว่าการจัดเก็บนี้ได้จากลำดับของการวาง Model ลงไปในไดอะแกรมจากการวาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 ตารางการเก็บข้อมูลจากโฟลว์ชาร์ต

L1	EndCondition	Stop	11	12	13	14	Yes	CodeEnd Condition	CodeStop
L2	Delay	Condition	3	4	5	6	Yes	Code Delay	CodeCondition
L3	Start	Delay	1	2	3	4	Yes	CodeStart	CodeDelay
L4	Condition	Input	5	6	7	8	Yes	CodeCon dition	CodeInput
L5	Condition	Output	5	6	9	10	No	CodeCon dition	CodeOutput
L6	Output	EndCondi tion	9	10	11	12	Yes	CodeOutput	CodeEndCondition
L7	Input	EndCondi tion	7	8	11	12	Yes	CodeInput	CodeEndCondition

หลังจากที่มีการนำข้อมูลเก็บในตารางแล้ว จะนำข้อมูลในตารางที่ 3.1 มาทำการจัดเรียงใหม่โดยมีการจัดเรียงลำดับโดยเริ่มที่โมเดล Start ตามด้วยโมเดลที่เชื่อมต่อกับโมเดล Start ไปเรื่อยๆ จนกระทั่งมีโมเดล Stop เป็นโมเดลสุดท้าย ดังตารางที่ 3.2

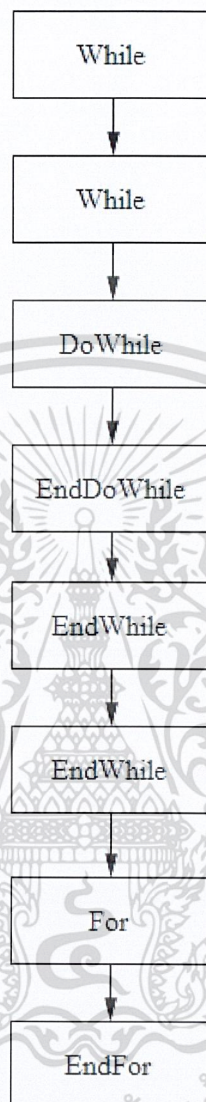
ตารางที่ 3.2 ตารางการเก็บข้อมูลหลังจากที่มีการจัดเรียงข้อมูลในตารางที่ 3.1

L3	Start	Delay	1	2	3	4	Yes	CodeStart	CodeDelay
L2	Delay	Condition	3	4	5	6	Yes	CodeDelay	CodeCondition
L4	Condition	Input	5	6	7	8	Yes	CodeCondition	CodeInput
L7	Input	EndCondition	7	8	11	12	Yes	CodeInput	CodeCondition
L5	Condition	Output	5	6	9	10	No	CodeCondition	CodeOutput
L6	Output	EndCondition	9	10	11	12	Yes	CodeOutput	CodeEndCon
L1	EndCondition	Stop	11	12	13	14	Yes	CodeEndCon	CodeStop

จากข้อมูลตารางที่ 3.1 และตารางที่ 3.2 จะทำการตรวจสอบข้อ 2 และ ข้อ 3 ซึ่งจะทำการตรวจสอบความถูกต้องของจำนวนเส้นเข้าและเส้นออกของแต่ละโมเดล และตรวจสอบว่าโมเดลวางถูกตำแหน่งหรือไม่ ดังตัวอย่างที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 3.2 ตัวอย่างการตรวจสอบว่าโมเดลวางถูกตำแหน่งหรือไม่โดยใช้สแตก ในตัวอย่างที่ 3.2 มีการยกตัวอย่างการวาดโฟลว์ชาร์ตเพียงบางส่วนดังรูปที่ 3.4



รูปที่ 3.4 ตัวอย่างบางส่วนของโฟลว์ชาร์ต

จากรูปที่ 3.4 นั้นจะทำการตรวจสอบว่าโมเดลวางถูกตำแหน่งหรือไม่โดยนำโมเดล While อันแรกนั้นใส่ลงสแตก จากนั้นจะเห็นได้ว่าโมเดลต่อไปคือโมเดล While อีกซึ่งจะเห็นได้ว่าโมเดล While ที่สองนั้นไม่สามารถปิดลูปได้จึงนำโมเดล While ใส่ลงสแตกอีกหนึ่งโมเดล จากนั้นจะพบโมเดล DoWhile จึงนำโมเดลนี้ใส่ลงสแตกอีก จะเห็นว่าขณะนี้ภายในสแตกนั้นมี 3 โมเดล ต่อไปจะเห็นว่าพบโมเดล EndDoWhile แต่โมเดล EndDoWhile นี้สามารถปิดลูปกับโมเดล DoWhile ได้ จึงนำโมเดล DoWhile ออกจากสแตก เช่นเดียวกันเมื่อพบโมเดล EndWhile โมเดลนี้ก็สามารถปิดลูปกับโมเดล While ได้ จึงนำโมเดล While ทั้งสองโมเดลออกจากสแตก สุดท้ายจะไม่มีโมเดลใดไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหลืออยู่ในสแตกแสดงว่าโมเดลทุกโมเดลในโฟลว์ชาร์ตนี้วางถูกตำแหน่งแล้ว หากโมเดลใดวางผิดตำแหน่งนั้นในการตรวจสอบของขั้นตอนนี้จะหยุดในทันทีแล้วทำการแสดงข้อความผิดพลาดให้ผู้ใช้ได้กลับไปแก้ไขต่อไปเมื่อการตรวจสอบความถูกต้องแล้ว ตารางดังกล่าวจะถูกนำไปใช้ในการตรวจสอบ Syntax และ generate code ต่อไป

## 2) การตรวจสอบ syntax

ในการตรวจสอบ Syntax นั้น ทำได้โดยใช้ข้อมูลจากตารางที่ 3.1 ตรวจสอบดูว่าโค้ดของโมเดลใดเป็น null หรือไม่ หากโมเดลมีโค้ดเป็น null จะมีข้อความเตือนข้อผิดพลาดว่าโมเดลใดยังไม่ มีโค้ดให้ผู้ใช้กลับไปแก้ไข

## 3) การ generate code

ในการ generate code มีการใช้ข้อมูลจากตารางที่ 3.2 โดยที่นำโค้ดของโมเดลตั้งแต่โมเดล Start จนถึงโค้ดของโมเดล Stop จะเห็นได้จากตารางที่ 3.2 ในการ generate code เริ่มจากโค้ดของโมเดล Start, โมเดล Delay, โมเดล Condition, โมเดล Input, โมเดล Condition (property ของเส้น เป็น No), โมเดล Output, โมเดล EndCondition และ โมเดล Stop

### 3.1.8 ทดสอบโปรแกรมและปรับปรุง

### 3.1.9 จัดทำเอกสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

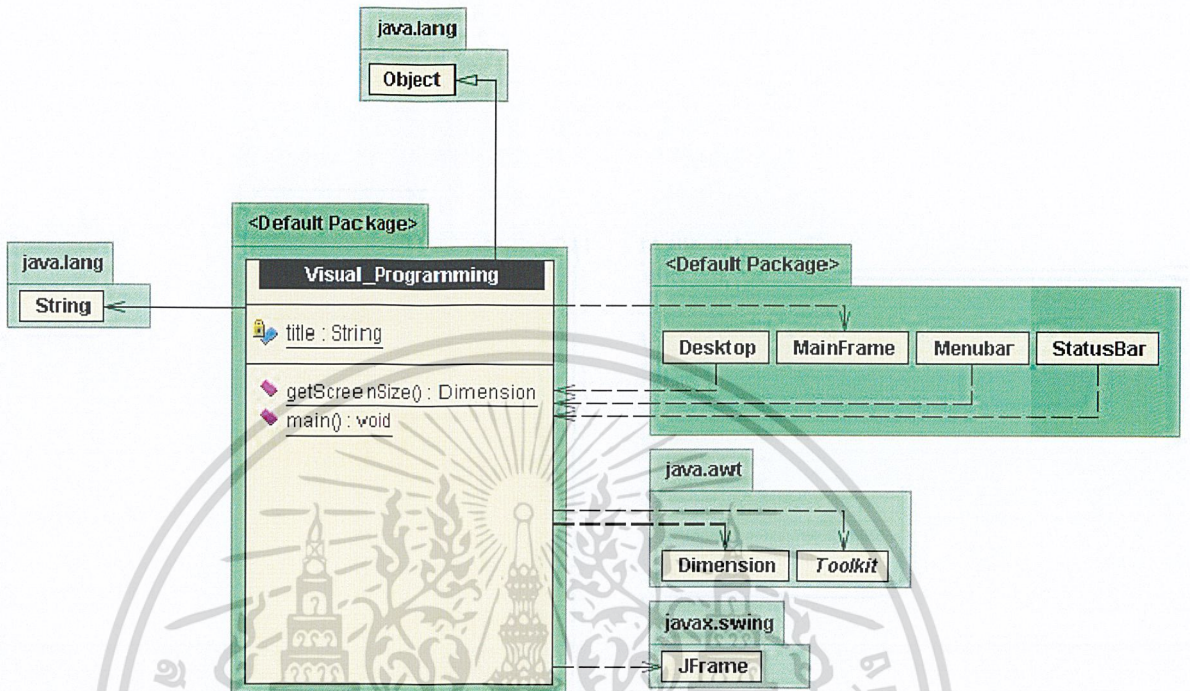
### 3.2 Class Diagram

โปรแกรม Visual Programming มีคลาสหลักๆ อยู่ 9 คลาส ดังนี้

- 1) **Mainframe** ทำหน้าที่สร้างและเป็นคอนเทรนเนอร์เก็บคอมโปเนนท์ทุกตัวในโปรแกรม รวมทั้งเป็นจุดศูนย์กลางในการส่งข้อความกันระหว่างคอมโปเนนท์ใด ๆ ด้วย
- 2) **MenuBar** ทำหน้าที่เก็บเมนูและเมนูไอเท็ม ซึ่งใช้เก็บคำสั่งการทำงานต่างๆ ของโปรแกรมรวมทั้งยังทำหน้าที่เป็นตัวสร้างทูลบาร์อีกด้วย
- 3) **ToolBar** ทำหน้าที่เป็นปุ่มของคำสั่งไว้บริการผู้ใช้ในการเข้าถึงคำสั่งการทำงานภายในเมนูให้รวดเร็วยิ่งขึ้น โดยจะถูกสร้างโดยคลาสเมนูบาร์
- 4) **StatusBar** ทำหน้าที่แสดงข้อความแนะนำ คำช่วยเหลือ บอกสถานะการทำงานของโปรแกรม เช่น โคออดิเนตของเมาส์บนไดอะแกรม และ สถานะของปุ่มเลือกต่างๆ ได้แก่ Num Lock, Cap Lock และ Scroll Lock
- 5) **Document** ทำหน้าที่แสดงรูปแบบของคำสั่งของแต่ละโมเดล เมื่อมีการเลือกโมเดลในไดอะแกรมตัวใดๆ ก็จะเรียกรูปแบบคำสั่งของโมเดลนั้นๆ มาแสดงที่หน้าต่างนี้ โดยข้อมูลที่จะนำมาแสดงนี้จะขึ้นอยู่กับคุณสมบัติของโมเดลนั้นๆ ด้วย คือเป็นลักษณะของไดนามิก
- 6) **Desktop** ทำหน้าที่เป็นคอนเทรนเนอร์เก็บไดอะแกรมทั้งหมด และทำหน้าที่ในการสร้างชุดคำสั่งภาษา C สำหรับไมโครคอนโทรลเลอร์ตระกูล MCS-51 ออกมาเก็บเป็นไฟล์เพื่อสามารถนำไปพัฒนาต่อไป
- 7) **Diagram** ทำหน้าที่ในการแสดงและเก็บโมเดลต่างๆ ที่ทำการวาดลงบนตัวมันเอง โดยจะมีลักษณะเป็น Internal Frame สามารถย่อ ขยาย เปิด ปิด หรือทำเป็นไอคอนได้
- 8) **AnyActionListener** ทำหน้าที่ในการเก็บค่าและส่งค่าต่อไปยัง CreateButton Model และเป็นส่วนที่ GenerateCode ส่งค่าไปยัง CreateButtonModel อีกด้วย
- 9) **CreatButton Model** ทำหน้าที่เก็บปุ่มคำสั่งที่ใช้ในการวาดโมเดลลงในไดอะแกรม เมื่อมีการเลือกคำสั่งในบาร์แล้ว ก็จะส่งค่าต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

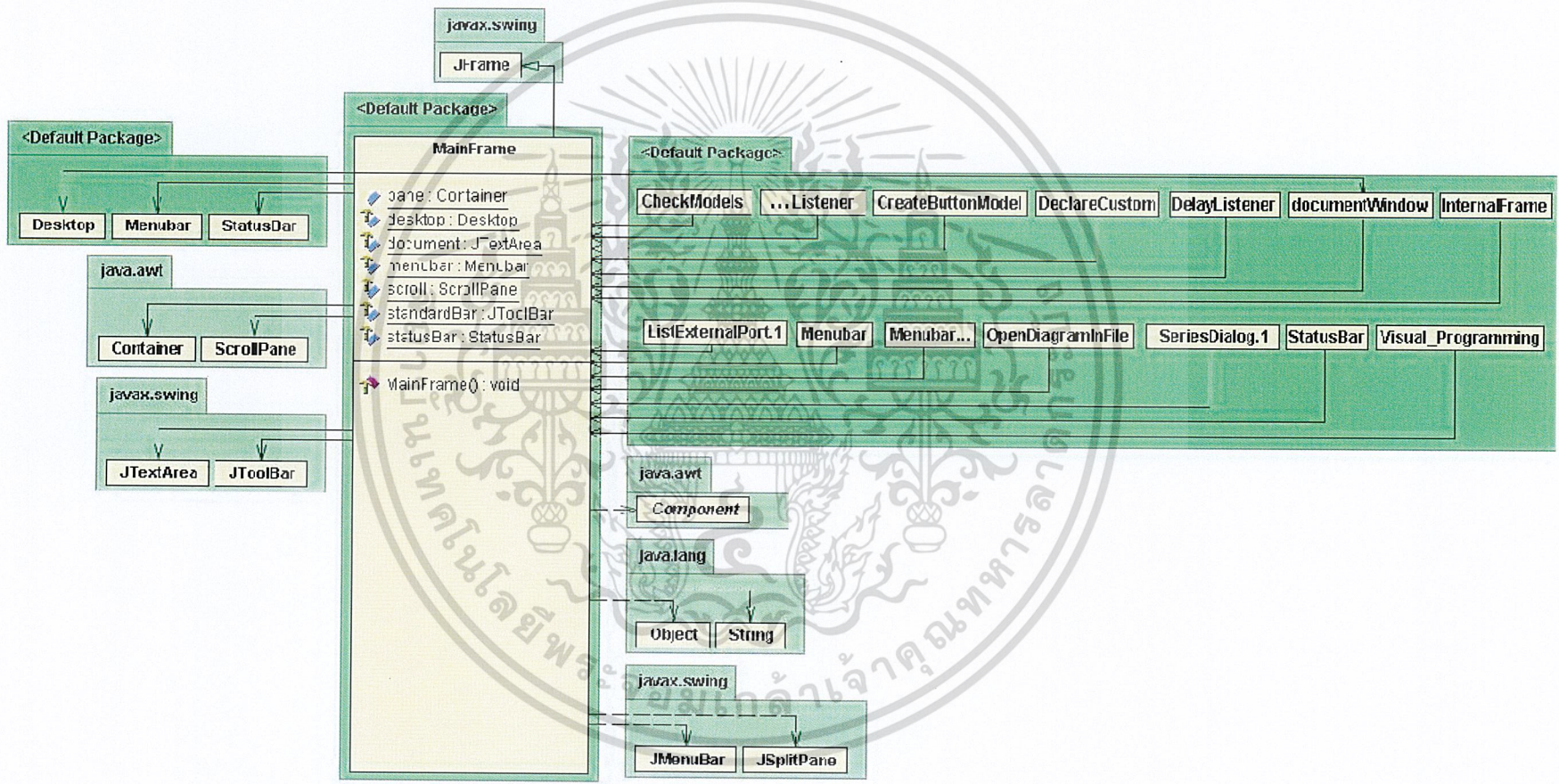
### 3.2.1 พัง Class Diagram ของคลาสต่างๆ



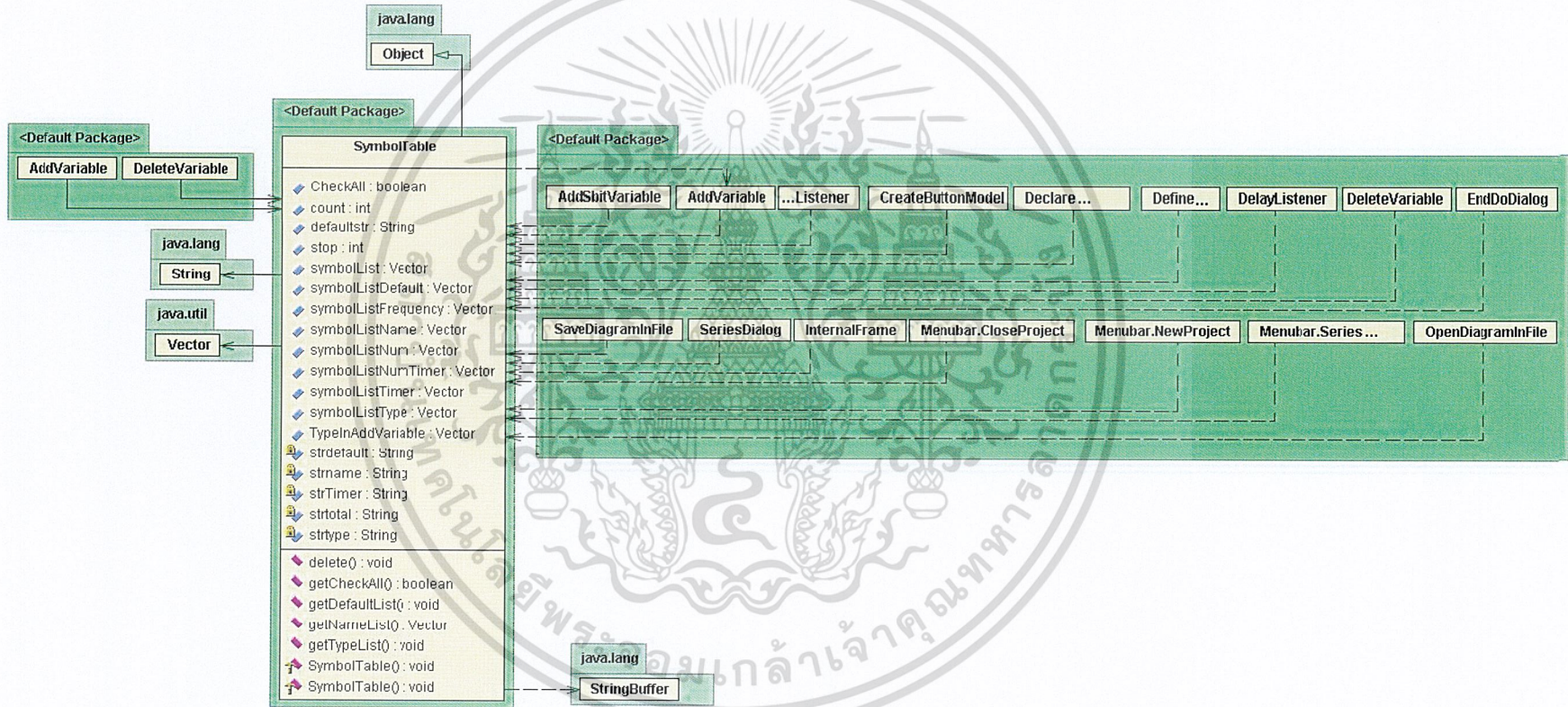
รูปที่ 3.5 คลาสหลัก Visual\_Programming

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

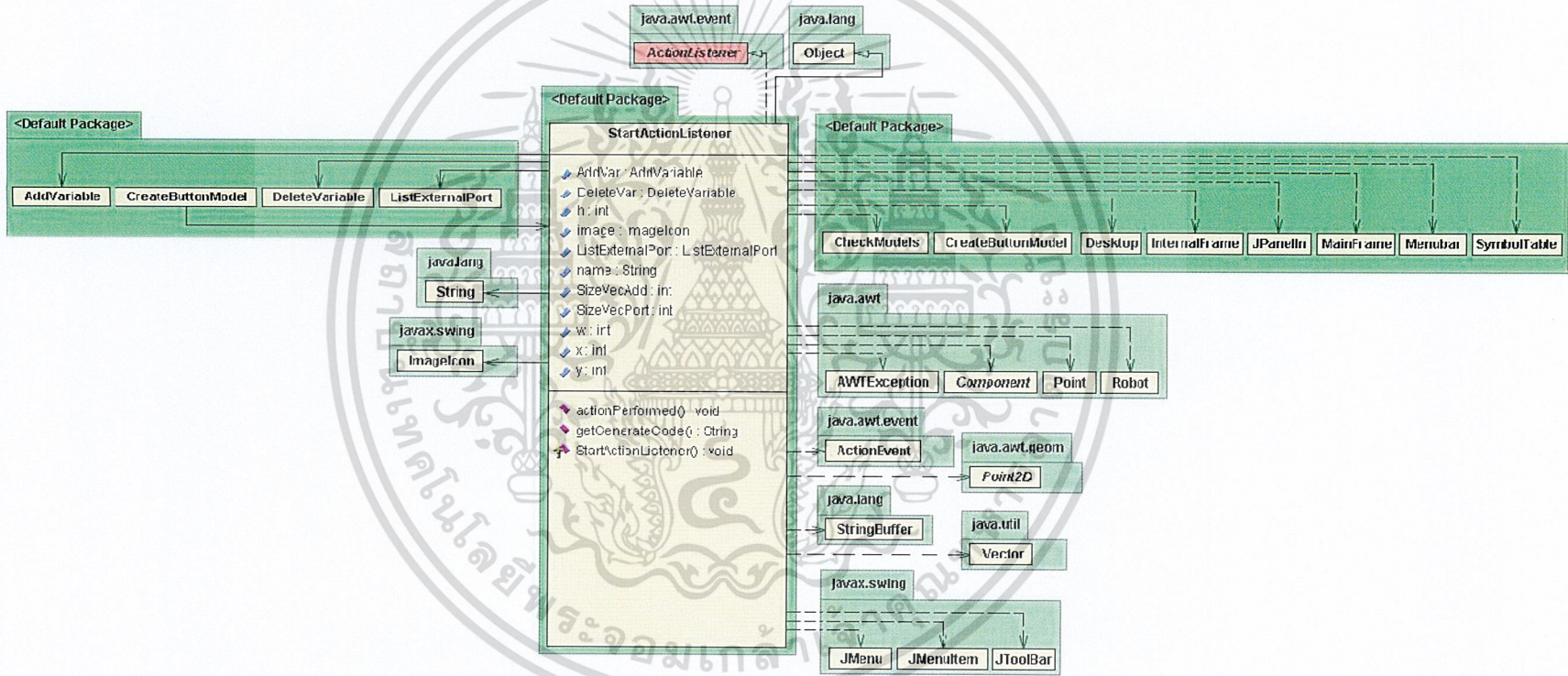
รูปที่ 3.6 กล้อง MainFrame



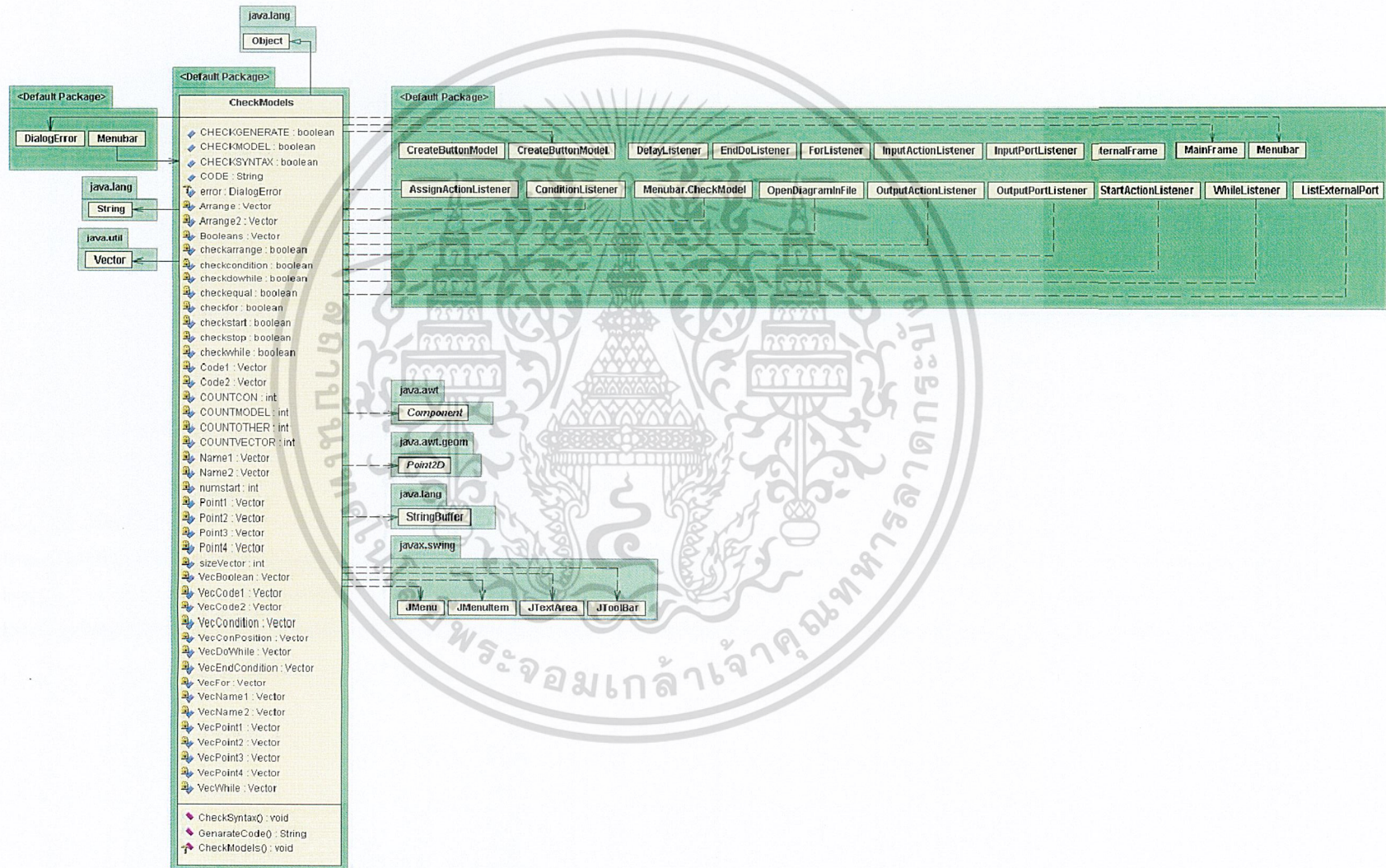
รูปที่ 3.7 กक्षा SymbolTable



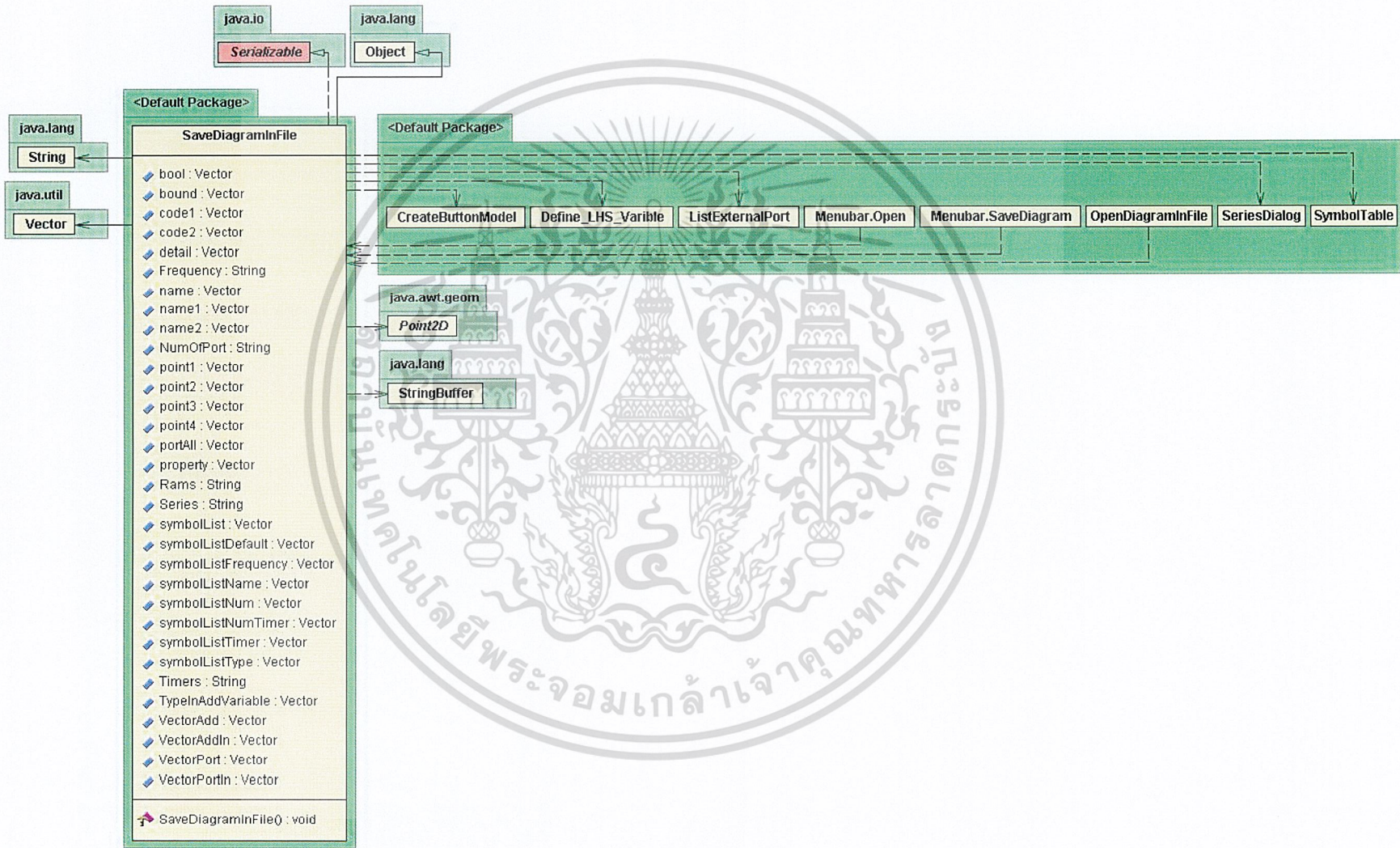
รูปที่ 3.8 คลาส StartActionListener



รูปที่ 3.9 คลาส CheckModels



รูปที่ 3.10 คลาส SaveDiagramInFile



## บทที่ 4

### ผลการทดลอง

#### 4.1 สถาปัตยกรรมของระบบและซอฟต์แวร์

##### 4.1.1 แพลตฟอร์มต่าง ๆ ที่ใช้

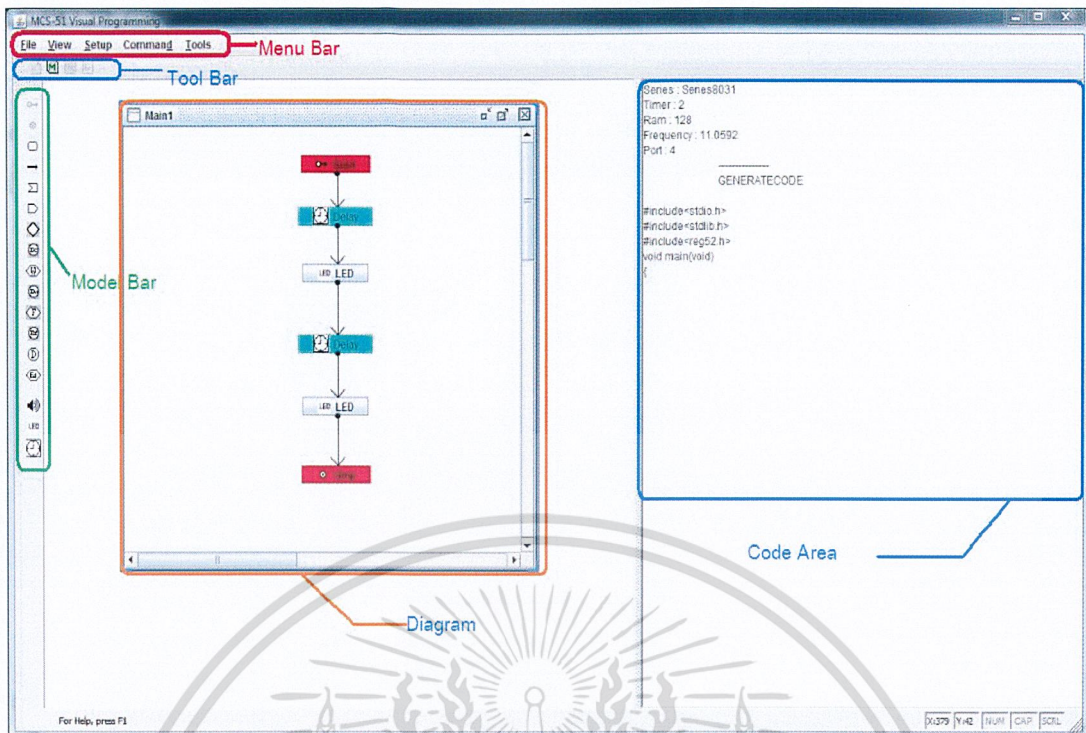
แพลตฟอร์มที่ใช้ประมวลผลโปรแกรม Visual Programming สามารถใช้แพลตฟอร์มใดก็ได้ เนื่องจากโปรแกรม Visual Programming นั้นถูกพัฒนาขึ้นด้วยภาษา Java เพราะ Java สามารถทำงานได้กับทุกสภาพแวดล้อม โดยเครื่องที่จะทำการประมวลผล Visual Programming จะต้องมีโปรแกรม Java Machine ( JVM ) ก็จะสามารถทำงานได้ ความต้องการของระบบ CPU ความเร็วอย่างต่ำ 166 MHz หน่วยความจำ RAM อย่างน้อย 32 เมกกะไบต์ หรือถ้าจะให้การประมวลผลเร็วขึ้นก็ควรเพิ่ม RAM มากเป็น 64 เมกกะไบต์ หรือมากกว่านั้น เพราะการประมวลผลโปรแกรมที่เป็น Graphics User Interface จะมีการใช้หน่วยความจำสูง Hard disk มีที่ว่างขนาดประมาณ 140 กิโลไบต์

##### 4.1.2 โครงสร้างของซอฟต์แวร์

โครงสร้างโปรแกรมเพื่อช่วยในการออกแบบและพัฒนาระบบ เครื่องมือช่วยพัฒนาโปรแกรมสำหรับไมโครคอนโทรลเลอร์ โครงสร้างของโปรแกรมแบ่งออกเป็น 7 ส่วนใหญ่ ๆ ได้แก่

- Menu Bar สำหรับเก็บคำสั่งการทำงานต่าง ๆ ของโปรแกรม
- Tool Bar สำหรับเก็บปุ่มคำสั่งที่มีการใช้งานบ่อย ๆ จากเมนูบาร์
- Status Bar สำหรับแสดงสถานะของโปรแกรม และข้อความช่วยเหลือต่าง ๆ
- Document สำหรับแสดงคำสั่งต่าง ๆ ของแต่ละโมเดล เมื่อมีการเลือกโมเดลนั้น ๆ
- Model Bar สำหรับเก็บปุ่มโมเดลต่าง ๆ ที่จะทำการวาดลงในไดอะแกรม
- Desktop สำหรับเก็บและทำงานเกี่ยวกับไดอะแกรมทั้งหมด
- Diagram สำหรับเก็บและแสดงโมเดลต่าง ๆ ที่ทำการวาดลงไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 GUI ของโปรแกรม Visual Programming สำหรับการสร้างโค้ด

### 1) Menu Bar

ส่วนของ Menu Bar ประกอบด้วย component ที่รองรับ 5 ส่วน ได้แก่

#### 1.1) File ประกอบด้วย

- New Project
- Open Diagram
- Save Code
- Save Diagram
- Close Project
- Exit

ในกรณีของการ Save Diagram นั้น ไม่ได้รวมถึงการ Save Code ด้วย หากต้องการ Save Code ต้องทำการ Save Code ต่างหาก

#### 1.2) View ประกอบด้วย

- Tool Bar
- Documentation
- Status Bar

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 1.3) Set up ประกอบด้วย

- Series board
- custom

1.4) Command เป็นส่วนที่ประกอบด้วยโมเดลที่นำไปใช้ในการวาดผังงาน เช่นเดียวกับใน ส่วนของ Model Bar ผู้ใช้สามารถเลือก Model ที่ต้องการใช้วาดลงในไดอะแกรมได้เช่นกัน

### 1.5) Tools ประกอบด้วย

- Check Model
- Check Syntax
- Generate Code

## 2) Tools Bar

ส่วนของ Tools Bar ประกอบด้วย component ที่รองรับ 4 ส่วน ได้แก่

- New project
- Check Model
- Check Syntax
- Generate Code

## 3) Model Bar ประกอบด้วย

- Start Model เป็นส่วนที่ผู้ใช้กำหนดตัวแปรและค่าที่ต้องการใช้มี 2 ส่วน คือ ส่วนที่เพิ่มและลบ
- Stop Model
- Assignment Model
- Condition Model
- End Condition Model
- While Model
- End While Model
- For Model
- End For Model
- Do While Model
- End Do While Model
- Buzzer Model
- LED Model
- Delay Model
- Number Model

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่หรือดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Relay Model

#### 4) ส่วนของพื้นที่ที่แสดงในส่วนของการ Generate Code

แสดงรายละเอียดในแต่ละ Model ดังนี้

- รายละเอียดเกี่ยวกับโมเดลนั้นๆ เช่น ชื่อ, รูปภาพ และพิกัดของโมเดล
- โค้ดที่ได้จากการ Generate ออกมาจากในแต่ละโมเดล
- หากโมเดลใดมีการเก็บค่า จะมีการแสดงค่าที่เก็บในแต่ละตัว เช่น ในโมเดล Start มีการเก็บค่า Variable Name ก็จะมีการแสดงค่านั้นในส่วนนี้

#### 5) พื้นที่ Diagram

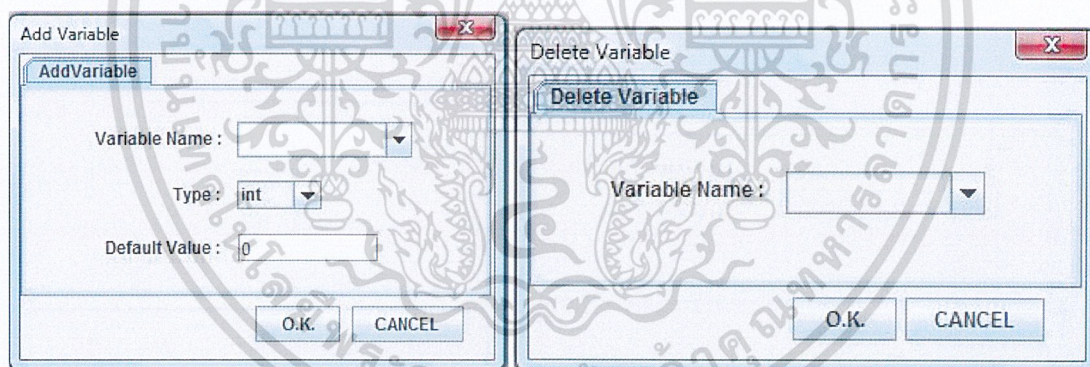
เป็นพื้นที่ที่ผู้ใช้วาดโฟลว์ชาร์ต

### 4.2 ลักษณะของ Pop up Menu ของ Model ต่างๆ

ในแต่ละโมเดลเมื่อผู้ใช้เลือกมาวาง ก็จะมี Pop up Menu ขึ้นมาให้ผู้ใช้กรอก Property ที่เกี่ยวข้องกับ Model นั้นลงไป และจะเก็บค่า Property เหล่านี้ไว้เพื่อสร้างโค้ดต่อไป

#### 1) Start Model

มี 2 Pop Up Menu คือ



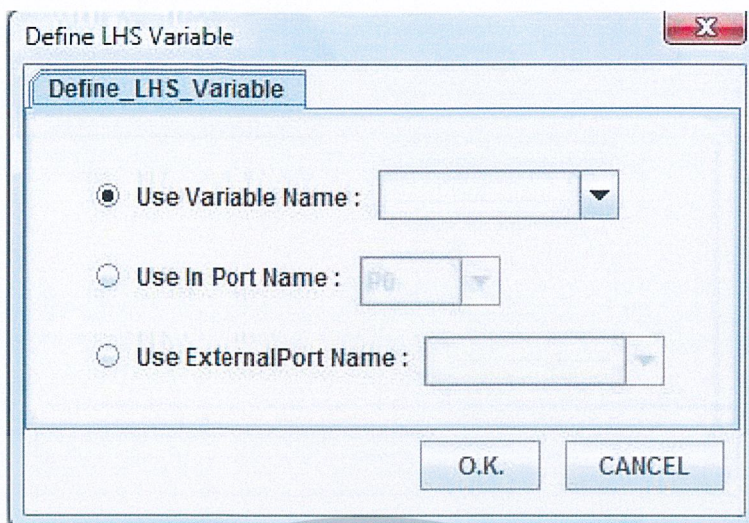
รูปที่ 4.2 Pop up Menu ของ Start Model

#### 2) Assignment Model

Pop up Menu ของ Assignment Model มี 2 ส่วนใหญ่ ๆ ได้แก่

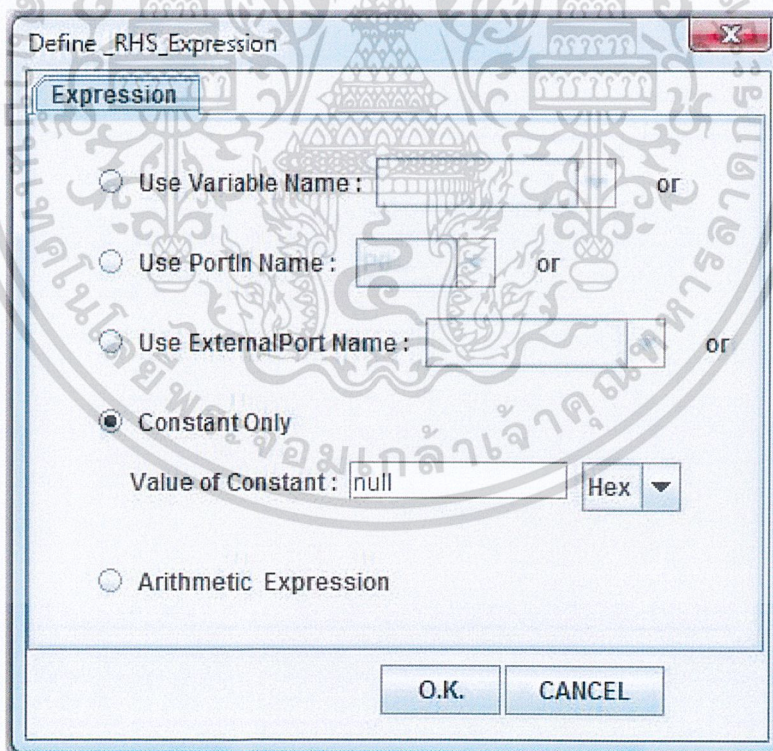
- การกำหนดค่าทางซ้ายของเครื่องหมาย
- การกำหนดค่าทางด้านขวาของเครื่องหมาย ซึ่งในการกำหนดค่าทางด้านขวานี้ สามารถเลือกกำหนดแบบของการกระทำทางคณิตศาสตร์ได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



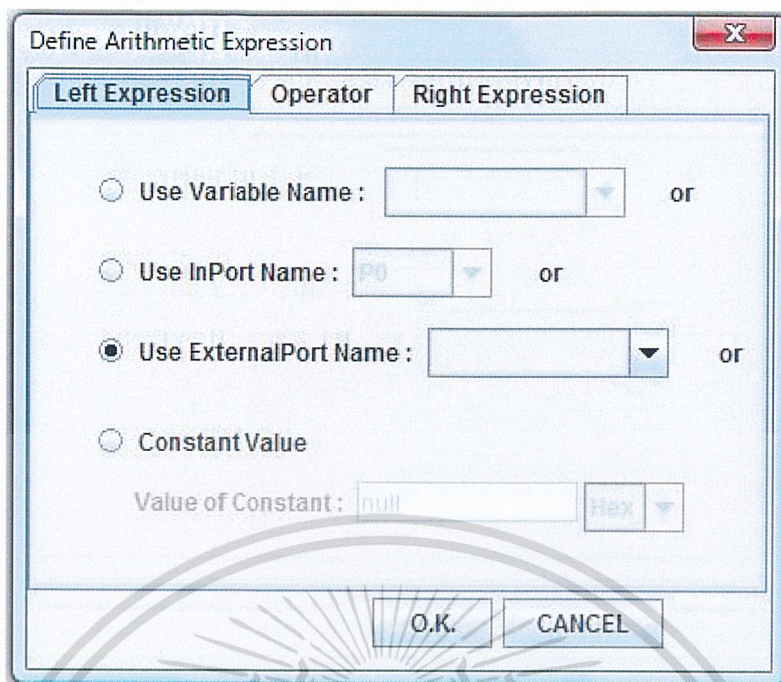
รูปที่ 4.3 Pop up การกำหนดค่าทางซ้ายของ Assignment Model

จากรูปที่ 4.3 จะเห็นได้ว่าค่าทางด้านซ้ายสามารถเลือกได้ 3 แบบ คือ เป็น ตัวแปร, พอร์ตภายในและ พอร์ตภายนอก แต่ต้องเลือกอย่างใดอย่างหนึ่งเท่านั้น ไม่สามารถเลือกพร้อมกันได้



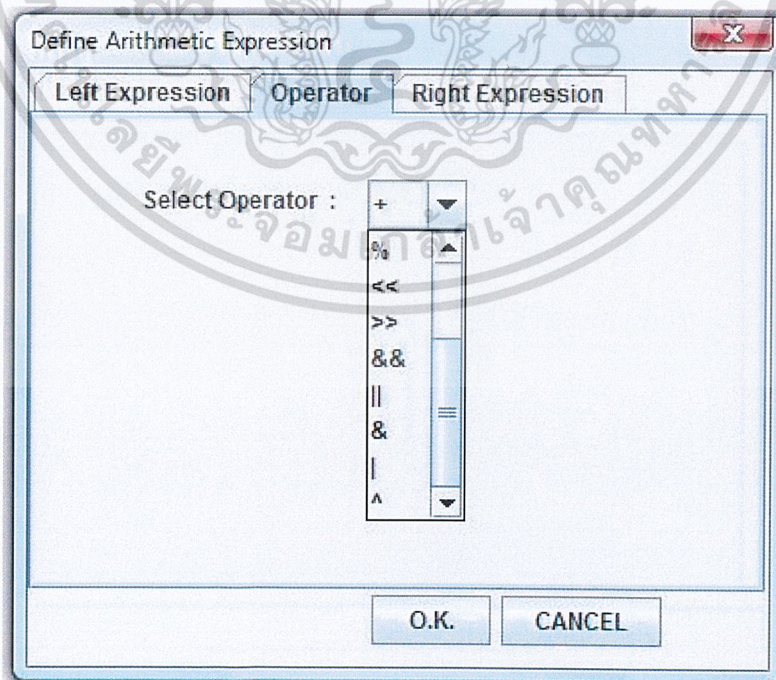
รูปที่ 4.4 Pop up Menu การกำหนดค่าทางด้านขวาของ Assignment Model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 การกำหนดค่าการกระทำทางคณิตศาสตร์ของ Assignment Model

จากรูปที่ 4.5 จะเห็นว่า มีทั้งการกำหนดค่าทางด้านซ้ายของ Expression , การกำหนดค่าทางด้านขวาของ Expression ซึ่งการกำหนดค่าทางด้านซ้ายและด้านขวานั้นเหมือนกัน และในส่วนของ Operator ดังจะแสดงในรูปที่ 4.6

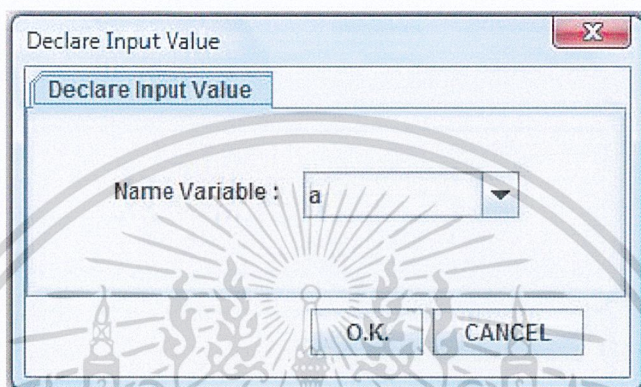


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
รูปที่ 4.6 Operator ของการกำหนดค่าการกระทำทางคณิตศาสตร์  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.6 จะเห็นได้ว่าการเพิ่ม Operator ที่เกี่ยวกับการใช้งานทางไมโครคอนโทรลเลอร์ได้แก่ <<, >>, &, &&, |, || และ ^

### 3) Input Model

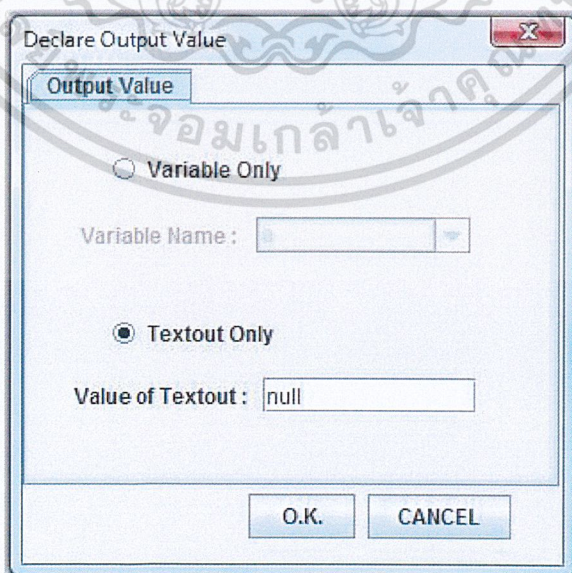
ในส่วนของ Input Model นั้นมีการรับค่าได้โดยระบุตัวแปรซึ่งประกาศไว้ใน Start Model ดังแสดงในรูปที่ 4.7



รูปที่ 4.7 Pop up Menu ของ Input Model

### 4) Output Model

กำหนดค่าของ output ได้ 2 แบบ คือ แบบระบุตัวแปร และแบบระบุข้อความดังแสดงในรูปที่ 4.8

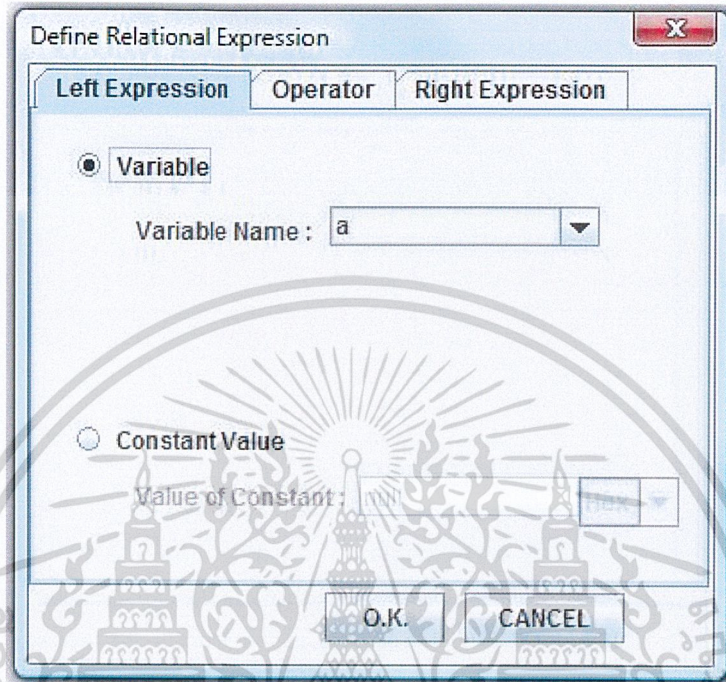


รูปที่ 4.8 Pop up Menu ของ Output Model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5) Condition Model

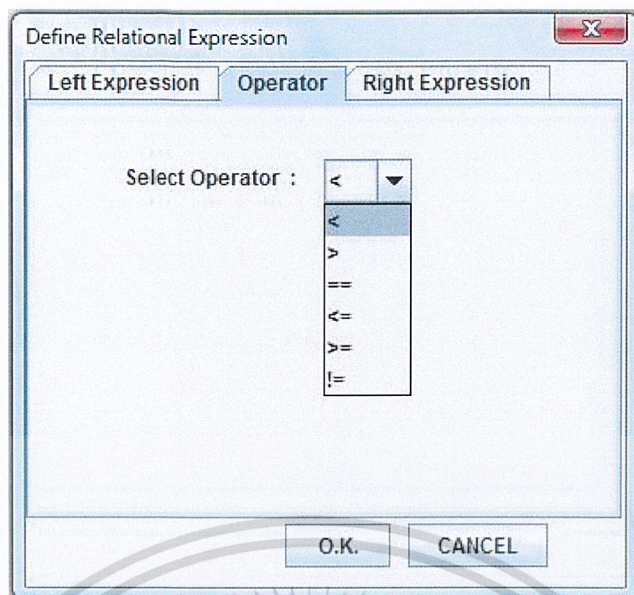
แบ่งเป็นการกำหนดค่าทางด้านซ้าย, การกำหนดค่าทางด้านขวา และการกำหนดโอเปอเรเตอร์ ดังแสดงในรูปที่ 4.9



รูปที่ 4.9 Pop up Menu ของ Condition Model โดยการกำหนดค่าทางด้านซ้าย

โดยที่การกำหนดค่าทางด้านซ้ายของ Expression และทางด้านขวาของ Expression ผู้ใช้สามารถป้อนค่าที่เป็นตัวแปร (Variable) หรือ ค่าคงที่ (Constant) โดยจะมี Operator ที่กั้นอยู่ระหว่าง expression ทางด้านซ้ายและขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

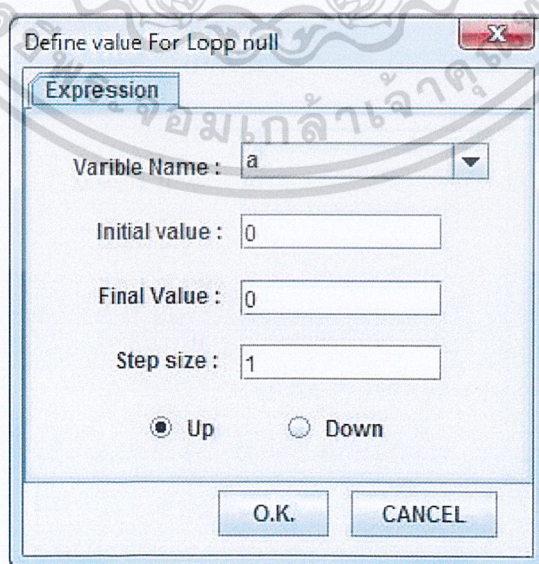


รูปที่ 4.10 Operator ของ Condition Model

ส่วน Pop up Menu ของ While Model, EndDoWhile Model มีรูปแบบเช่นเดียวกับ Condition Model จึงไม่ขอแสดง ณ ที่นี้

#### 6) For Model

โดยที่ Pop up menu ของ For Model นี้ผู้ใช้งานจะต้องกำหนดค่าของรายละเอียดทั้งหมด ได้แก่ ชื่อตัวแปร (Variable), ค่าเริ่มต้น, ค่าสุดท้าย และลักษณะของลูป (ใช้สำหรับการนับขึ้น / นับลงเท่านั้น)

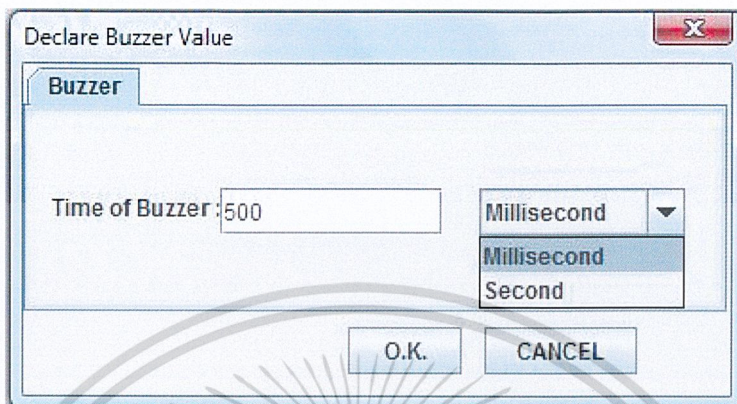


รูปที่ 4.11 Pop up Menu ของ For Model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 7) Buzzer Model

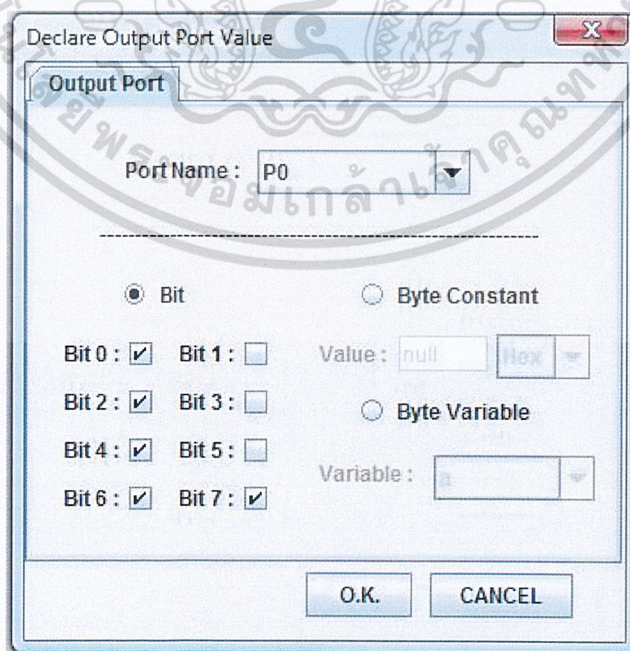
จากรูปที่ 4.12 ในส่วนของ Buzzer Model นี้ ผู้ใช้จะต้องระบุเวลาของ buzzer ซึ่งสามารถเลือกเป็นหน่วยวินาทีหรือมิลลิวินาที



รูปที่ 4.12 Pop up Menu ของ Buzzer Model

## 8) LED Model

ในส่วนนี้ผู้ใช้จะต้องกำหนดค่าของโมเดล LED นั้น โดยต้องระบุพอร์ตที่ติดต่อกับไมโครคอนโทรลเลอร์ เมื่อทำการเลือกแล้ว ผู้ใช้สามารถกำหนดระหว่าง Bit กับ Byte Constant ได้อย่างใดอย่างหนึ่งเท่านั้น ดังรูปที่ 4.13

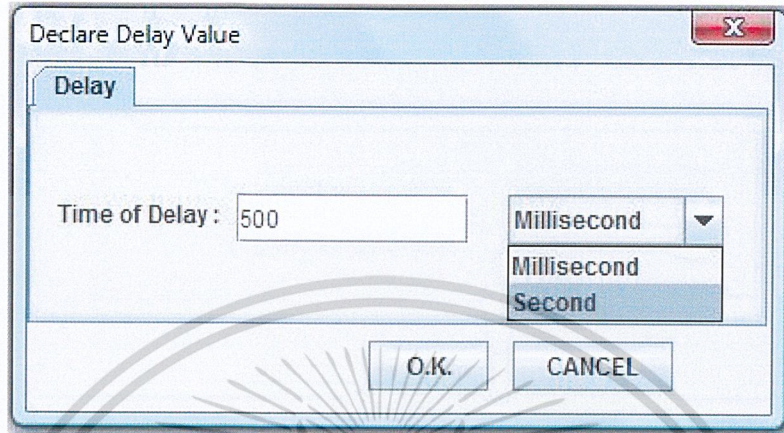


รูปที่ 4.13 Pop up Menu ของ LED Model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้นำเอกสารทุกครั้งที่มีการนำไปใช้

### 9) Delay Model

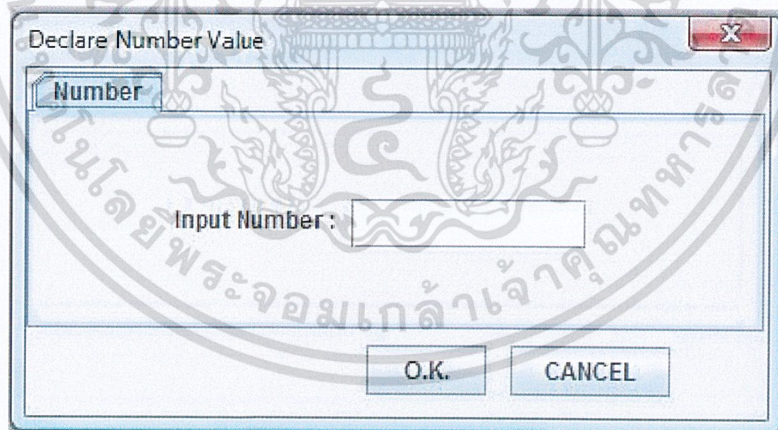
กำหนดค่าของโมเดล Delay นั้นจะต้องระบุเวลาของดีเลย์ โดยสามารถเลือกหน่วยเป็น มิลลิวินาทีหรือวินาทีได้



รูปที่ 4.14 Pop up Menu ของ Delay Model

### 10) Number Model

ในส่วนของ Number Model ต้องระบุตัวเลข ตั้งแต่ 0-9 โดยสามารถใส่ได้สูงสุด 8 หลัก

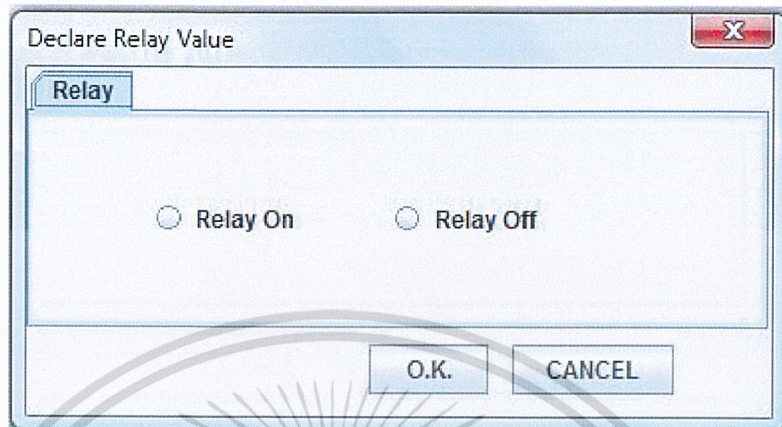


รูปที่ 4.15 Pop up Menu ของ Number Model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 11) Relay Model

ในส่วนของ Relay Model นั้น ต้องเลือกระหว่าง on กับ off



รูปที่ 4.16 Pop up Menu ของ Relay Model

### 4.3 Property ที่สำคัญของแต่ละ Component ของไดอะแกรม

ในที่นี้ จะกล่าวถึง Property ที่สำคัญของแต่ละ Component ของไดอะแกรม Property เหล่านี้จะใช้ในการสร้างโค้ดที่สมบูรณ์ Component ในที่นี้ ได้แก่ For, While, DoWhile, Condition, Assignment, Input, Output, Input port, Output port และ Delay

- 1) For Model ประกอบด้วย แอททริบิวต์ได้แก่
  - ตัวแปร ที่ใช้ เป็น Index
  - ค่าเริ่มต้น เป็นค่าคงที่
  - ค่าสิ้นสุด เป็นค่าคงที่
  - Step size
  - Up/Down
- 2) While Model ,Do While และ IF ประกอบด้วย แอททริบิวต์ได้แก่
  - Relational Expression อย่างง่าย ประกอบด้วย
    - ตัวแปร ด้านซ้ายหรือค่าคงที่
    - เครื่องหมาย >, <, >=, <=, !=, ==
    - ตัวแปรด้านขวาหรือค่าคงที่
- 3) Input ประกอบด้วย แอททริบิวต์ได้แก่
  - ตัวแปร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4) Output ประกอบด้วย แอทรินิวต์ได้แก่
  - ตัวแปร
  - ข้อความ
- 5) Assignment ประกอบด้วย แอทรินิวต์ได้แก่
  - ตัวแปรด้านซ้ายของเครื่องหมาย Assignment
  - ค่าคงที่ ตัวแปรด้านขวาหรือ Arithmetic Expression อย่างง่าย ประกอบด้วย
    - ตัวแปร ด้านซ้ายหรือค่าคงที่
    - เครื่องหมาย +, -, \*, /, %, <<, >>, &, &&, |, || และ ^
- 6) Buzzer ประกอบด้วย แอทรินิวต์ ได้แก่
  - เวลาของการหน่วงเวลา
  - หน่วยของการหน่วงเวลา
- 7) LED ประกอบด้วยแอทรินิวต์ ได้แก่
  - พอร์ต
  - การเข้าถึงแบบบิต
  - การเข้าถึงแบบไบต์
  - ตัวแปร
- 8) Delay ประกอบด้วยแอทรินิวต์ ได้แก่
  - เวลาของการหน่วงเวลา
  - หน่วยของการหน่วงเวลา
- 9) Number ประกอบด้วยแอทรินิวต์ ได้แก่
  - ตัวเลข 0-9 ใส่ได้มากที่สุด 8 หลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### ผลการดำเนินงาน ข้อจำกัด และข้อเสนอแนะ

#### 5.1 ผลการดำเนินงาน

เครื่องมือช่วยพัฒนาโปรแกรมสำหรับไมโครคอนโทรลเลอร์ ในปัจจุบันมีความสามารถ ดังนี้

1. นักพัฒนาโปรแกรมสามารถสร้างโปรแกรมภาษา C สำหรับไมโครคอนโทรลเลอร์ได้อย่างอัตโนมัติ โดยไม่ต้องเขียนด้วยมือ
2. นักพัฒนาโปรแกรมสามารถนำโค้ดที่ได้จากเครื่องมือช่วยพัฒนาโปรแกรมสำหรับไมโครคอนโทรลเลอร์ไปใช้ได้จริง
3. นักพัฒนาโปรแกรมสามารถบันทึก (save) โค้ดและไฟล์ชาร์ตได้ เพื่อนำมาใช้ต่อไปได้ ทั้งนี้ยังสามารถเปิดงานเก่าเพื่อใช้งานหรือแก้ไขเพิ่มเติมได้
4. โปรแกรมมีการแจ้งเตือนเมื่อนักพัฒนาโปรแกรมสร้างไฟล์ชาร์ตผิด เช่น ไม่มี end loop , ไม่มีเส้นเข้า-ออกไคอะแกรม, ไม่มีค่าในไคอะแกรม เป็นต้น ทำให้ผู้ใช้รู้ข้อผิดพลาดและกลับไปแก้ไขให้ถูกต้องได้

#### 5.2 ข้อจำกัด

เครื่องมือช่วยพัฒนาโปรแกรมสำหรับไมโครคอนโทรลเลอร์ มีข้อจำกัดดังนี้

1. ในการกำหนดรุ่นของไมโครคอนโทรลเลอร์ และพอร์ตภายนอกนั้นเป็นของตระกูล MCS-51ชิปเบอร์ MPC82G516Aเท่านั้น ไม่สามารถใช้กับไมโครคอนโทรลเลอร์ตระกูลอื่น และเบอร์อื่นๆได้
2. ในการวาดไฟล์ชาร์ตของ Condition Model นั้นจะต้องทำการลากเส้นออก 2 เส้นเสมอ ไม่มีเส้นเชื่อมอัตโนมัติ
3. ในการวาด Loop ประเภทต่างๆ ไม่สามารถวาดเป็น cycle ได้
4. ไม่รองรับตัวแปรประเภทอะเรย์
5. For loop ใช้ได้เป็น Counter Loop เท่านั้น
6. ไม่สามารถสร้างฟังก์ชันได้ การทำงานของโปรแกรมจะทำในเมนหลักเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.3 ข้อเสนอแนะ

เครื่องมือช่วยพัฒนาโปรแกรมสำหรับไมโครคอนโทรลเลอร์ อาจจะไปทำการปรับปรุงได้ดังนี้

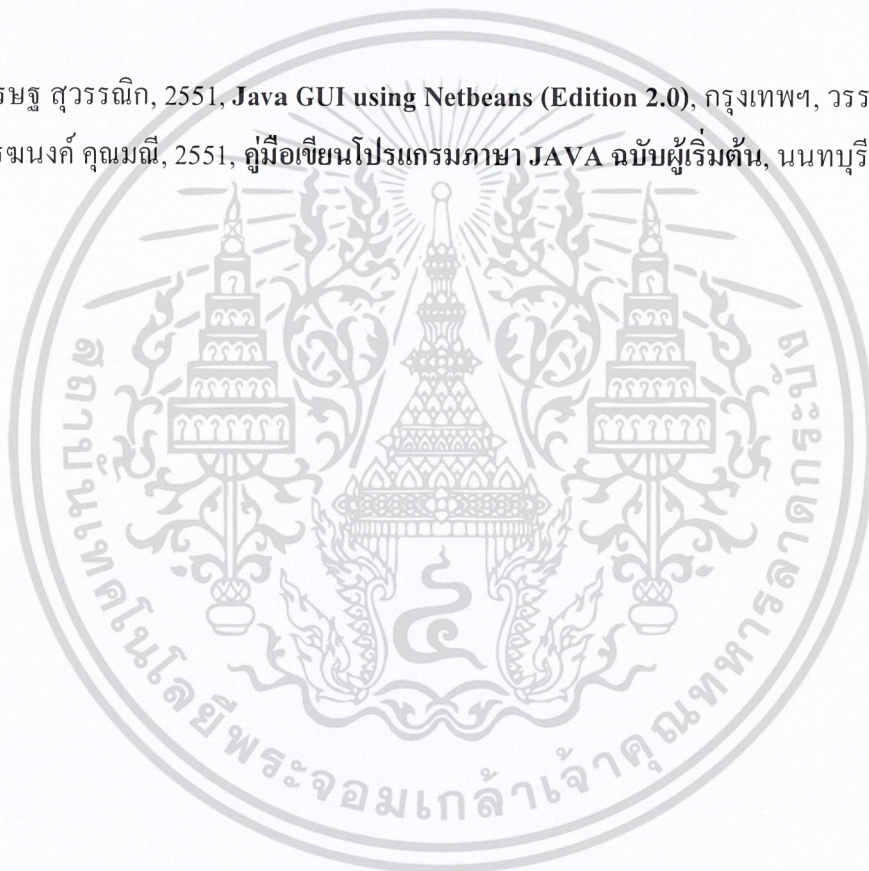
1. เปลี่ยนแปลงหรือเพิ่มเติมการ Generate Code โดย Generate เป็นภาษาอื่นได้ในบางคอมไพเลอร์เช่น ภาษา Assembly เพื่อเพิ่มขีดความสามารถการ Generate Code
2. ปรับปรุงให้สร้างโฟลว์ชาร์ตอัตโนมัติโดยไม่ต้องให้ผู้ใช้เสียเวลาในการลากเส้นเชื่อมต่อ
3. ปรับปรุงให้รองรับการทำงานกับอะเรย์อย่างง่ายเพื่อความสะดวกในการเขียนโค้ด
4. ปรับปรุงความสามารถของการเขียน For loop
5. ปรับปรุงให้โปรแกรมสามารถใช้งานได้กับไมโครคอนโทรลเลอร์เบอร์อื่นๆเพิ่มเติมได้
6. เพิ่ม Help ในซอฟต์แวร์เพื่อเป็นระบบช่วยเหลือและแนะนำวิธีการใช้งานแก่ผู้ใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## เอกสารอ้างอิง

- [1] ชีรวัดน์ ประกอบผล, 2545, การพัฒนาไมโครคอนโทรลเลอร์ด้วยภาษาซี, กรุงเทพฯ, สำนักงานสมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), หน้า 35-136.
- [2] ชีรวัดน์ ประกอบผล, 2544, การประยุกต์ใช้งานไมโครคอนโทรลเลอร์, ครั้งที่ 5, กรุงเทพฯ, สำนักงานสมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), หน้า 19-91.
- [3] ชีรวัดน์ ประกอบผล, 2553, คู่มือการเขียนโปรแกรม ภาษา Java, กรุงเทพฯ, ชิมพลีฟาย
- [4] วรณิกา เนตรงาม, 2545, คู่มือการเขียนโปรแกรมภาษา JAVA ฉบับผู้เริ่มต้น, นนทบุรี, อินโฟเพรส
- [5] วรเศรษฐ สุวรรณิก, 2551, Java GUI using Netbeans (Edition 2.0), กรุงเทพฯ, วรณิก
- [6] อนรรฆนงค์ คุณมณี, 2551, คู่มือเขียนโปรแกรมภาษา JAVA ฉบับผู้เริ่มต้น, นนทบุรี, ไอดีซีฯ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

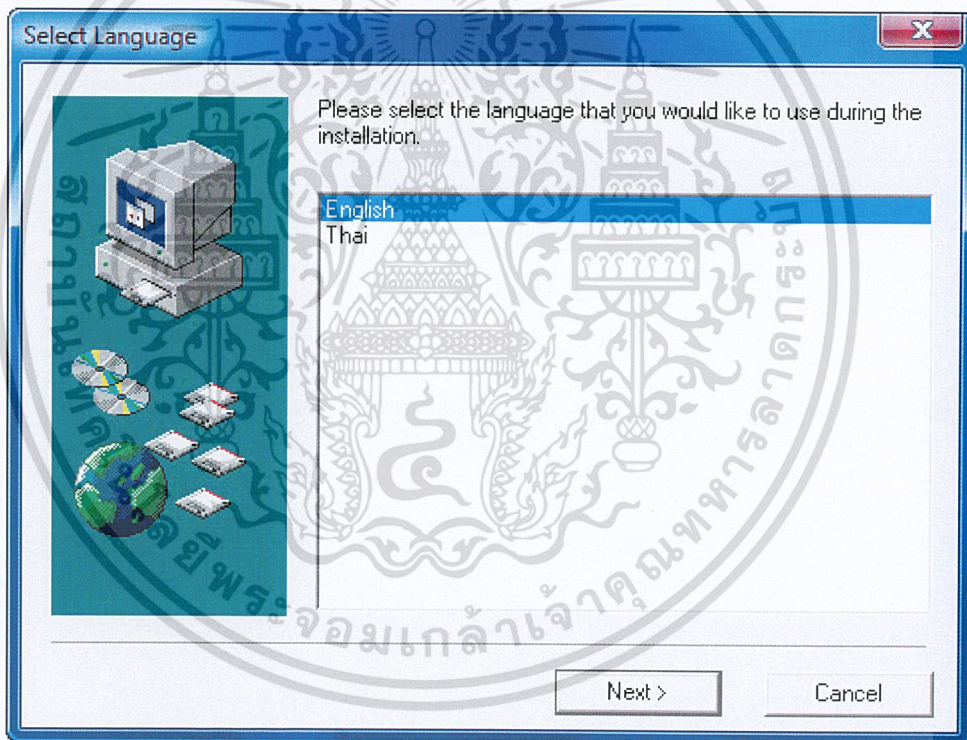
## ภาคผนวก ก

### คู่มือการติดตั้งโปรแกรม

ในการติดตั้งโปรแกรม Visual Programming มีขั้นตอนดังต่อไปนี้

1. ดับเบิ้ลคลิกที่ไอคอน setup จากนั้นโปรแกรม Installer จะทำงานและเริ่มกระบวนการติดตั้งเป็นขั้นตอน ดังต่อไปนี้

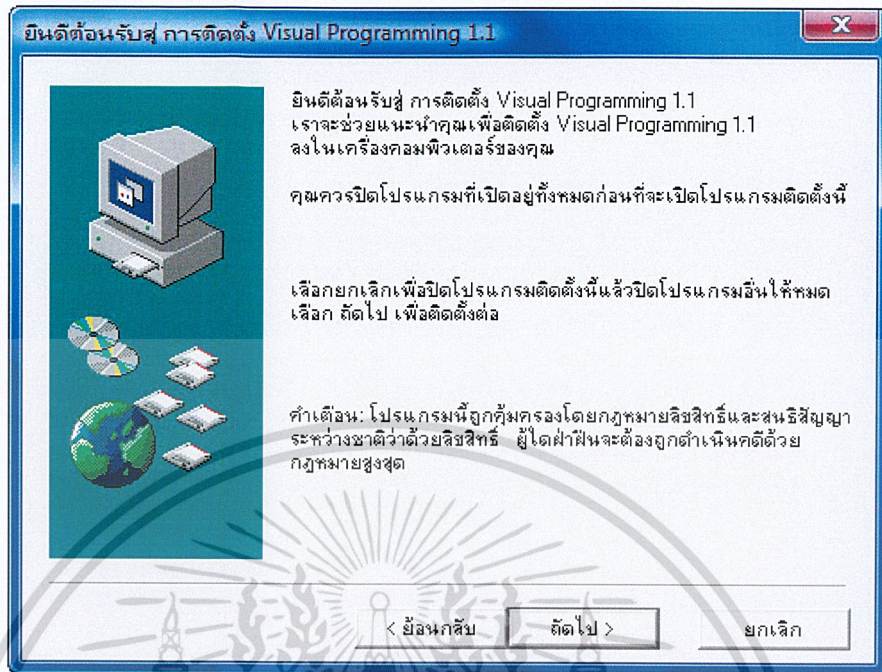
- 1.1 โปรแกรม Installer แสดงทางเลือกการเลือกภาษา โดยมีภาษาอังกฤษซึ่งเป็นค่า default และภาษาไทย คลิกที่ ถัดไป เพื่อไปหน้าต่อไป



รูปที่ ก.1 หน้าจอ โปรแกรมติดตั้งแสดงทางเลือกของภาษา

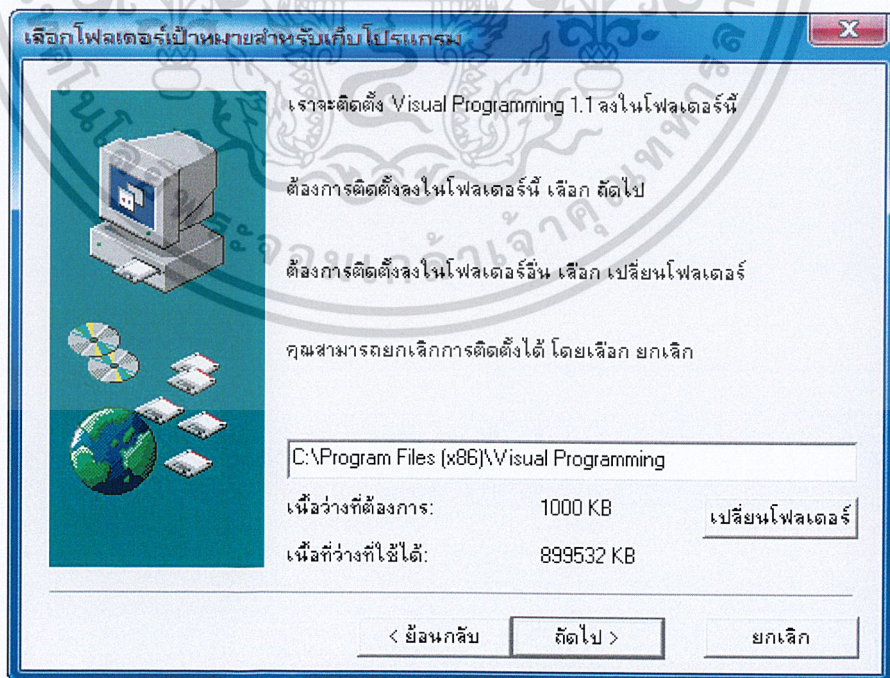
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.2 โปรแกรม Installer แสดงข้อความต้อนรับ คลิกที่ ถัดไป เพื่อ ไปหน้าต่อไป



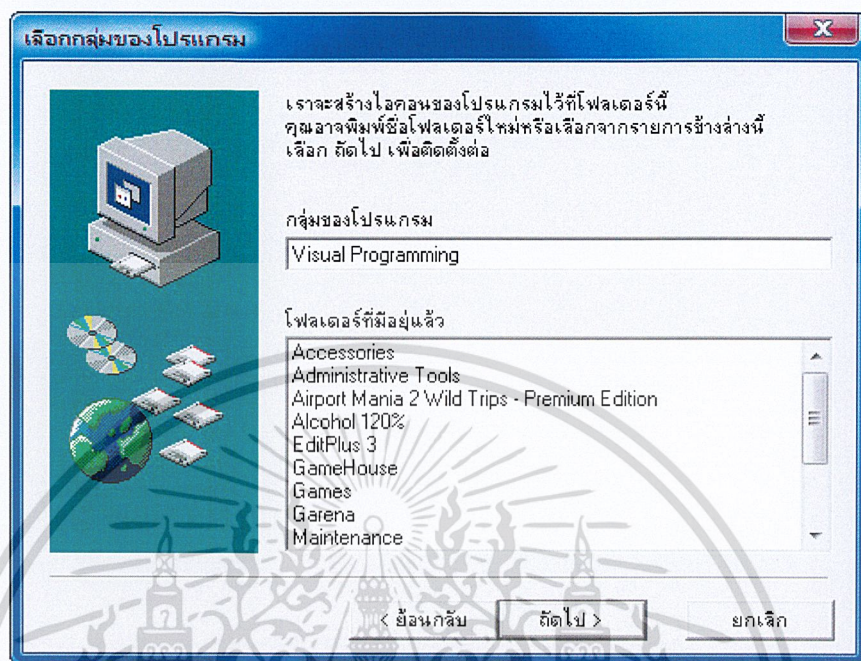
รูปที่ ก.2 หน้าจอ โปรแกรมติดตั้งแสดงข้อความต้อนรับ

## 1.3 ในหน้าจอถัดไป แสดงทางเลือกในการติดตั้งโปรแกรมในเครื่อง โดยสามารถเลือกที่ตั้งได้โดยคลิกที่ เปลี่ยนโฟลเดอร์ แล้วคลิก ถัดไป



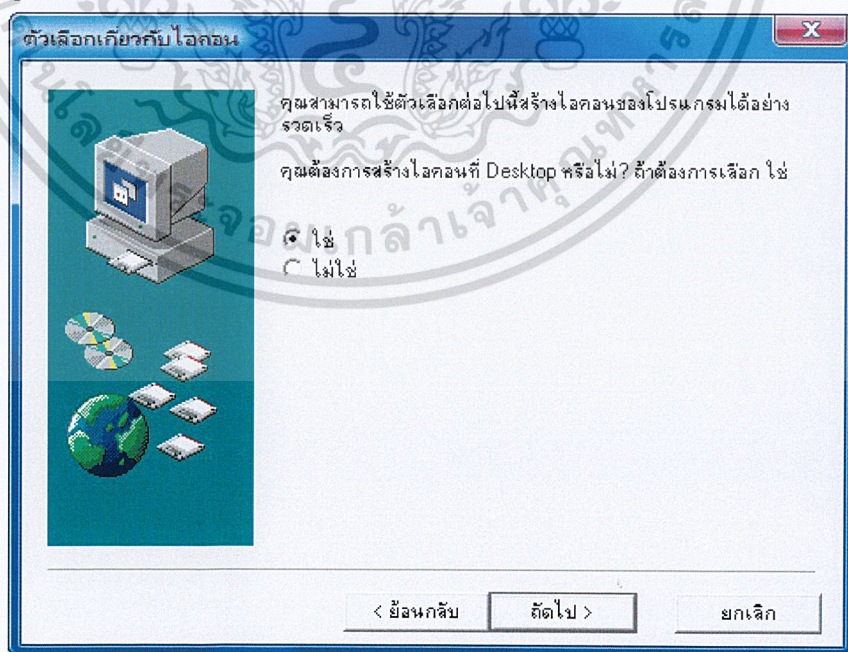
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ ก.3 หน้าจอ โปรแกรมติดตั้งแสดงที่เก็บโปรแกรม  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ในหน้าจอถัดไป แสดงทางเลือกในการสร้างโฟลเดอร์ใหม่พร้อมตั้งชื่อโฟลเดอร์ หรือตั้งค่าให้งานอยู่ในโฟลเดอร์ที่มีอยู่แล้วโดยใช้เก็บไอคอนของโปรแกรมแล้วคลิก ถัดไป



รูปที่ ก.4 หน้าจอ โปรแกรมติดตั้งแสดงกลุ่มของ โปรแกรม

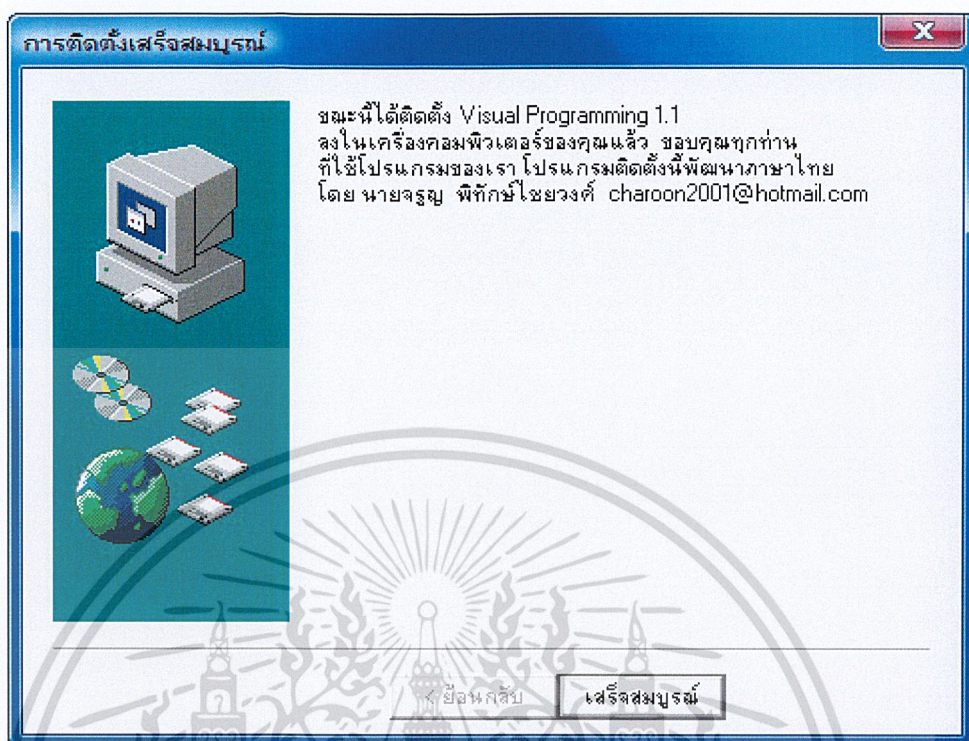
1.5 หน้าจอแสดงตัวเลือกเกี่ยวกับ ไอคอน โดยแสดงทางเลือกในการสร้างไอคอนที่ Desktop เมื่อเลือกแล้วคลิก ถัดไป



รูปที่ ก.5 หน้าจอ โปรแกรมติดตั้งแสดงตัวเลือกเกี่ยวกับ ไอคอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในกรณีฉุกเฉินเท่านั้น ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.6 หน้าจอแสดงข้อความการติดตั้งที่เสร็จสมบูรณ์ แล้วคลิก เสร็จสมบูรณ์



รูปที่ ก.6 หน้าจอโปรแกรมติดตั้งแสดงข้อความการติดตั้งเสร็จสมบูรณ์

## 2. รันโปรแกรม Visual Programming



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะในชั้นเรียนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ข

### คู่มือการใช้งานโปรแกรม

#### 1. เริ่มต้นขั้นตอนการใช้งานโปรแกรม Visual Programming

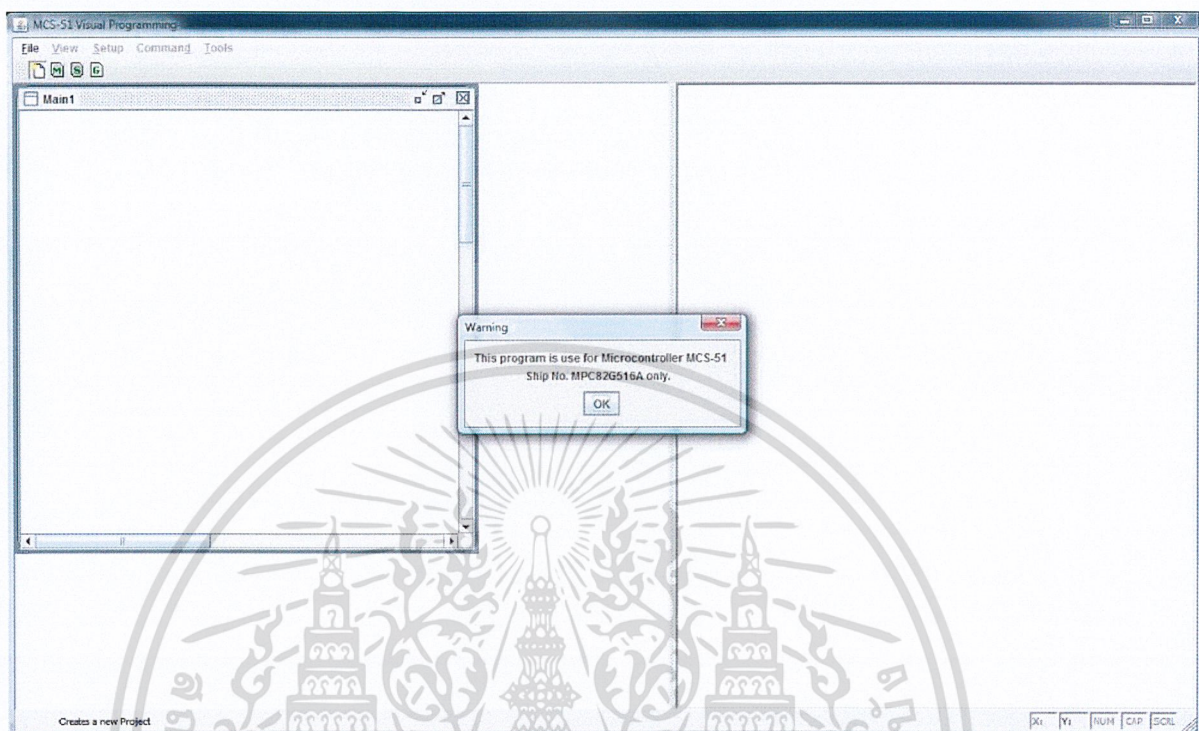
เริ่มต้นการใช้งานโปรแกรม Visual Programming เมื่อเข้าสู่โปรแกรมแล้ว จะปรากฏหน้าจอตามรูปที่ ข.1



รูปที่ ข.1 หน้าจอเริ่มต้น โปรแกรม Visual Programming

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นกดที่ปุ่ม New หรือกดที่ File>>New Project จะมี Main1 ปรากฏพร้อมข้อความแจ้งเตือนว่า  
This program is use for Microcontroller MCS-51 Ship No. MPC82G516A only.

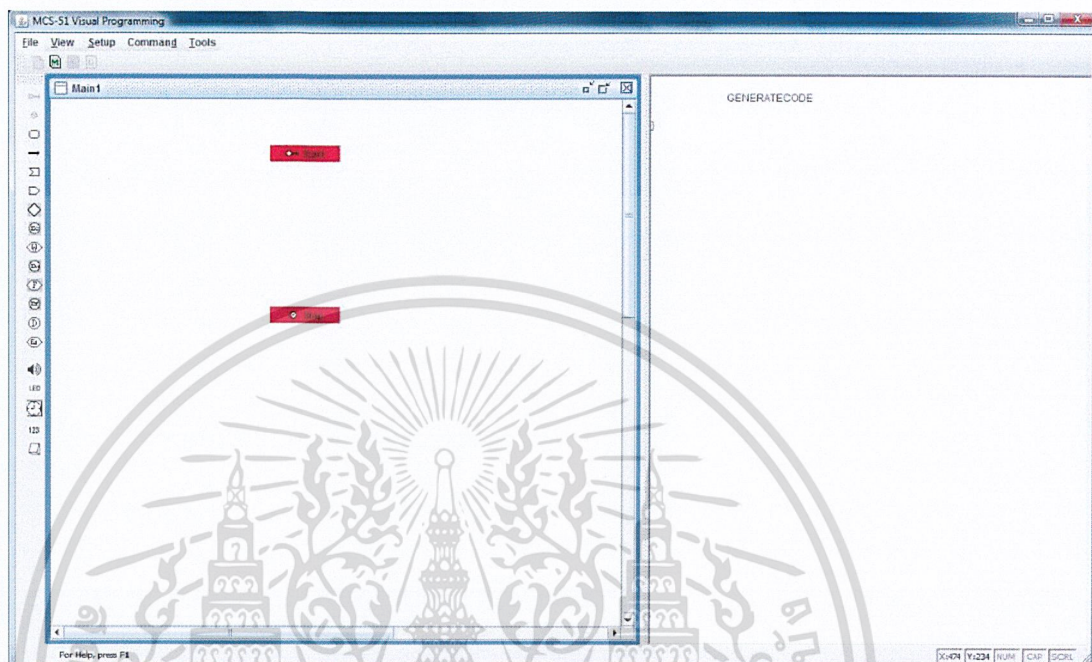


รูปที่ ข.2 หน้าจอแรกเมื่อมีการเปิด New Project ขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2. การวาดผังงานลงในโปรแกรม Visual Programming

การวาดผังงานลงในโปรแกรม Visual Programming ทำได้โดยการเลือกโมเดลแล้วคลิกที่ตัวโมเดลที่ต้องการเลือกใช้ ตัวโมเดลที่วางลงบนไดอะแกรมแล้วสามารถเลื่อนที่ได้



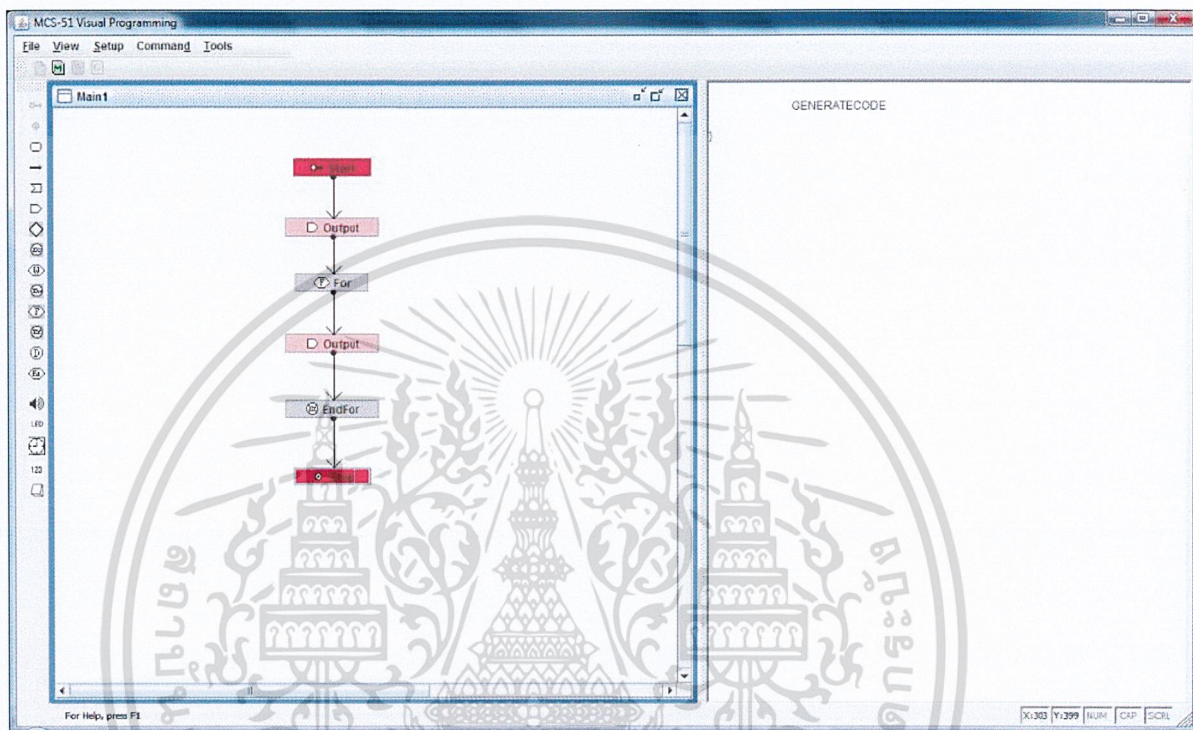
รูปที่ ข.3 การวาดผังงาน

จากรูปที่ ข.3 ในการวาดผังงานลงบนไดอะแกรมนั้นสำหรับโมเดล Start และ Stop นั้นสามารถใช้ได้ครั้งเดียวเท่านั้น ไม่สามารถลบโมเดลออกได้ เพราะในการ generate code จำเป็นจะต้องมี Start และ Stop เสมอ ในการวาดผังงานนั้นสามารถลากเส้นหรือกำหนดค่าข้อมูลของผังงานก่อนก็ได้ ในที่นี้จะขออธิบายในขั้นตอนของการลากเส้นก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3. การวาดเส้นเชื่อมของผังงาน

ในการวาดเส้นเชื่อมของผังงานสามารถทำได้โดย กดที่ รูปลูกศรตรง Model Bar หรือ Command >> Flow แล้วทำการคลิกในตัวของโมเดลเริ่มต้นที่ต้องการทำการวาดเส้น จากนั้นทำการ drag mouse ไปยังตัวโมเดลที่ต้องการสิ้นสุดการวาดเส้น หากผู้ใช้ลากเส้นผิดพลาดสามารถลบเส้นเชื่อมได้โดยการนำเมาส์ไปคลิกที่เส้นแล้วลบออก



รูปที่ ข.4 การวาดเส้นของผังงาน

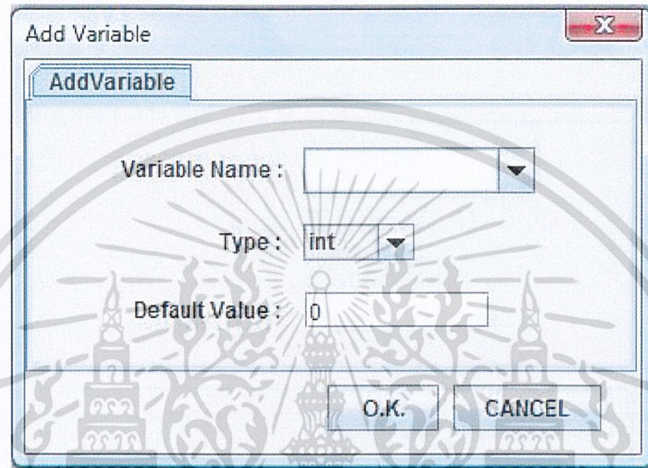
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4. การระบุข้อมูลของระบบ

เราสามารถระบุข้อมูลของฟังก์ชันที่ได้วางลงในโคแอดแกรม ได้โดยการกดเมาส์ปุ่มด้านขวา เมื่อกดปุ่มด้านขวาของเมาส์แล้วในแต่ละโมเดลที่สามารถกำหนดค่าได้นั้นจะมี pop up ขึ้นมาให้ผู้ใช้ได้กำหนดค่าลงไป

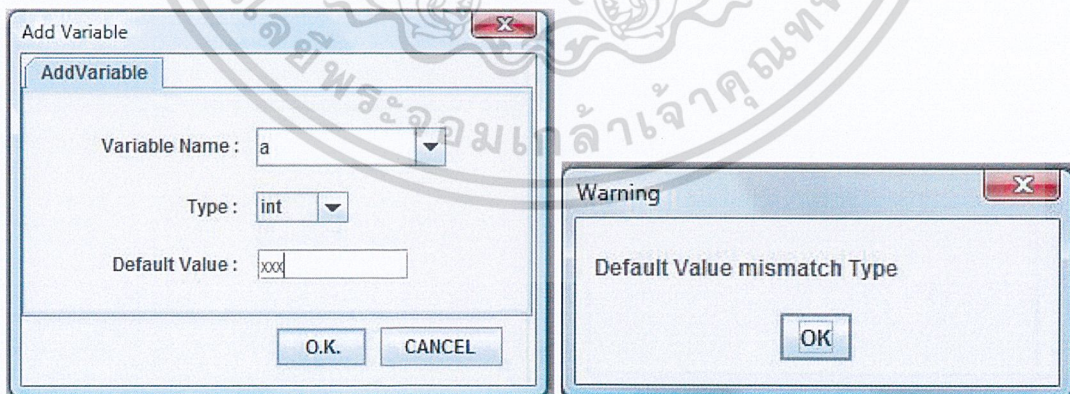
##### 4.1 โมเดล Start

ผู้ใช้สามารถทำการเพิ่มตัวแปรและลบตัวแปรได้ที่โมเดล Start โดยที่การเพิ่มตัวแปรผู้ใช้จะต้องระบุข้อมูลดังรูปที่ ข.5



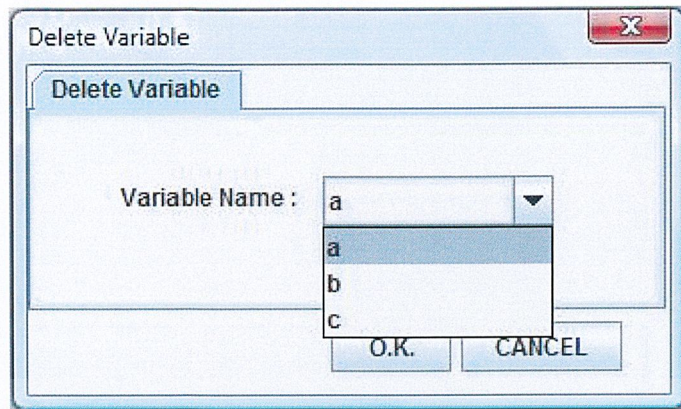
รูปที่ ข.5 การระบุค่าของโมเดล Start

จากรูปที่ ข.5 จะต้องระบุค่าของชื่อตัวแปร (VariableName), ชนิดของตัวแปร (Type), การกำหนดค่า (Default Value) หากมีการกำหนดค่าผิดจะขึ้นข้อผิดพลาด ดังรูปที่ ข.6



รูปที่ ข.6 การแสดงข้อความผิดพลาดเมื่อผู้ใช้กรอกข้อมูลของฟังก์ชัน ไม่ถูกต้อง

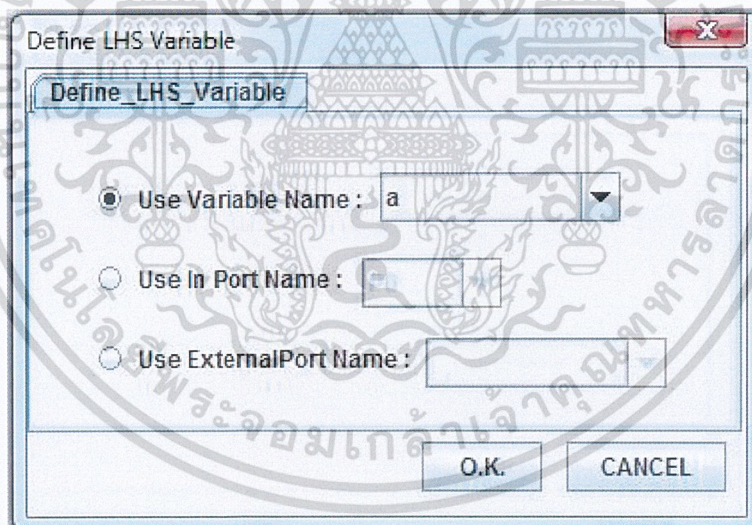
ส่วนการลบตัวแปรนั้นผู้ใช้จะต้องทำการเลือกตัวแปรที่ต้องการลบ ดังรูปที่ ข.7 ซึ่งตัวแปรที่สามารถลบได้นั้นจะต้องไม่มีการใช้ตัวแปรนั้นในโมเดลใดๆ ก็ตาม



รูปที่ ข.7 Pop up Menu ของการลบตัวแปร

#### 4.2 โมเดล Assignment

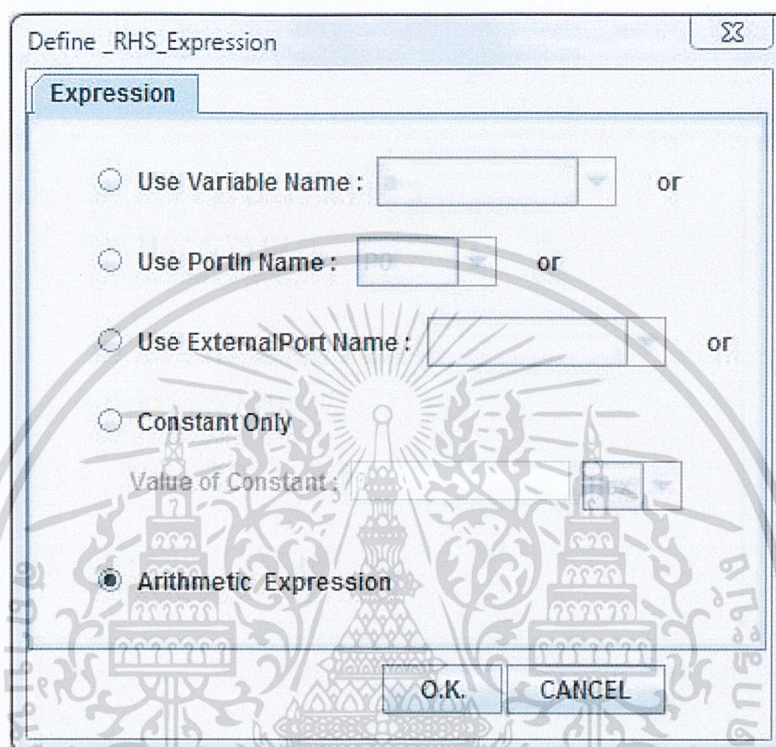
ภายในโมเดลผู้ใช้งานจำเป็นต้องระบุค่าของ Expression ทางซ้ายและ Expression ทางขวา โดยที่ Expression ทางซ้ายนั้น ผู้ใช้จะต้องเลือกกำหนดค่าเป็นตัวแปร, พอร์ตภายใน หรือพอร์ตภายนอก(หากมีการกำหนด) ดังรูปที่ ข.8



รูปที่ ข.8 Pop up Menu ของการกำหนดค่าทางด้านซ้ายของ Expression

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

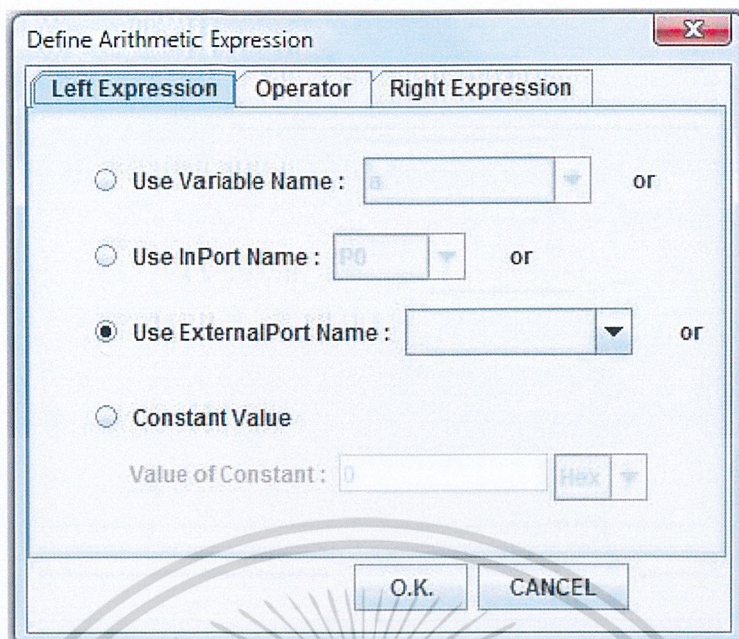
ส่วนการกำหนดค่าทางด้านขวาของ Expression นั้น ผู้ใช้สามารถเลือกกำหนดข้อมูลเป็น ชื่อตัวแปร, พอร์ตภายใน, พอร์ตภายนอก, ค่าคงที่ ซึ่งในที่นี้จำเป็นต้องระบุหน่วยของค่าคงที่ด้วยมีให้เลือก 2 หน่วยคือ Hex หรือ Decimal หรือหากผู้ใช้ต้องการกำหนดค่าแบบการกระทำทางคณิตศาสตร์นั้นก็สามารถทำได้โดยการเลือก Arithmetic Expression



รูปที่ ข.9 Pop up Menu ของการกำหนดค่าทางขวาของ Expression

การกำหนดค่าทางคณิตศาสตร์นั้นผู้ใช้จะต้องกำหนดค่าทั้งด้านซ้ายและทางด้านขวา ทั้งนี้ผู้ใช้สามารถเลือกกำหนดค่าเป็นแบบชื่อตัวแปร, พอร์ตภายใน, พอร์ตภายนอก (หากมีการกำหนด), ค่าคงที่ซึ่งค่าคงที่ที่ต้องระบุหน่วยเช่นเดียวกัน เหมือนกันทั้งทางด้านซ้ายและด้านขวา นอกจากนี้ผู้ใช้ยังจำเป็นต้องกำหนด Operator ด้วย โดยการคลิกที่ Operator จะมีเครื่องหมายของการกระทำที่ผู้ใช้ได้ทำการเลือก

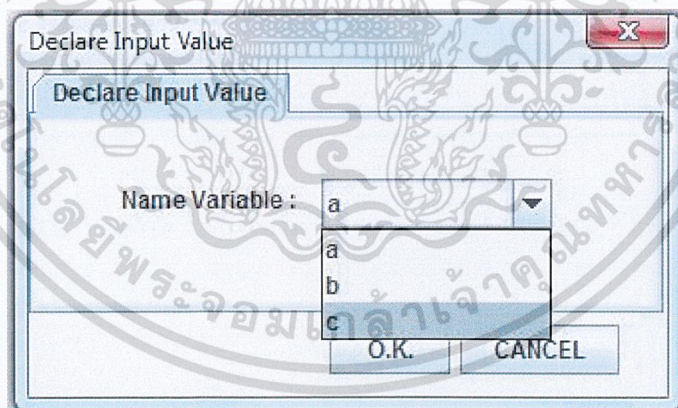
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.10 Pop up Menu ที่ใช้กำหนดค่าของการกระทำทางคณิตศาสตร์

#### 4.3 โมเดล Input

หากผู้ใช้มีการเลือกใช้โมเดล Input นั้นผู้ใช้จะต้องทำการเลือกตัวแปรที่ต้องการใช้ได้ที่ Combobox list หลังจากนั้นก็กดปุ่ม OK ดังรูปที่ ข.11

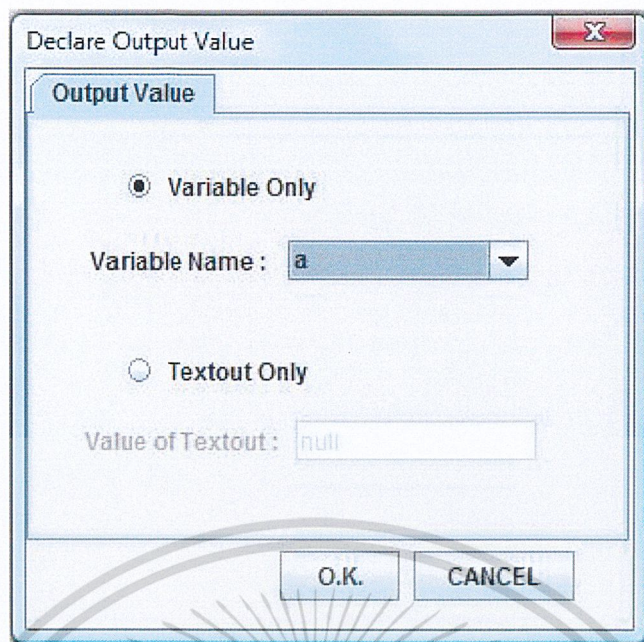


รูปที่ ข.11 Pop up Menu ที่ทำการกำหนดค่าของโมเดล Input

#### 4.4 โมเดล Output

โมเดล Output นี้ผู้ใช้สามารถเลือกกำหนดค่าในรูปแบบของชื่อตัวแปรโดยที่ผู้ใช้สามารถเลือกใช้ตัวแปรที่ต้องการในส่วนของ Variable Name หรือเป็นแบบข้อความโดยที่ผู้ใช้สามารถเลือกได้ในส่วนของ Textout Only ดังรูปที่ ข.12

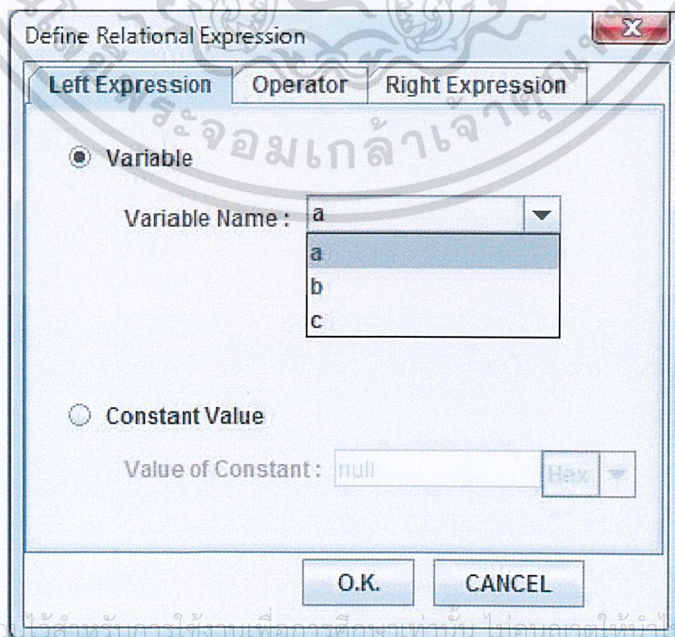
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.12 Pop up Menu ที่ทำการกำหนดค่าของโมเดล Output

#### 4.5 โมเดล Condition

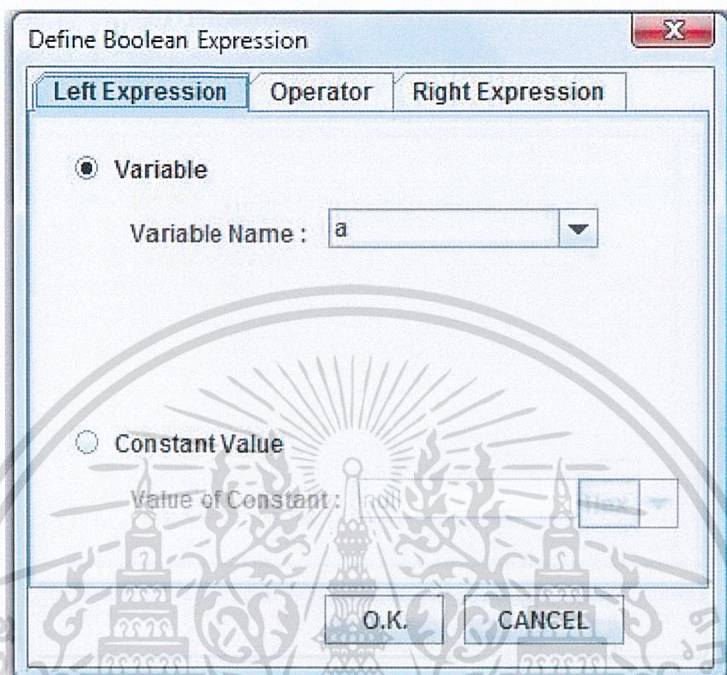
การกำหนดค่าของโมเดล Condition นี้จะต้องมีการกำหนดค่าทั้งทางด้านซ้ายและทางด้านขวาของ Expression โดยที่การกำหนดค่าทางด้านซ้ายและทางด้านขวามี property ที่ให้ผู้ใช้ได้เลือกเหมือนกันคือสามารถเลือกใช้เป็นชื่อตัวแปรหรือค่าคงที่ โดยการกำหนดค่าทางด้านซ้ายทำได้โดยการเลือกที่ pane ของ Left Expression ส่วนการกำหนดค่าทางด้านขวาทำได้โดยการเลือกที่ pane ของ Right Expression จากนั้นผู้ใช้จะต้องกำหนด operator ทำได้โดยการเลือกที่ pane ของ Operator



รูปที่ ข.13 Pop up Menu ที่ทำการกำหนดค่าของโมเดล Condition

#### 4.5 โมเดล While

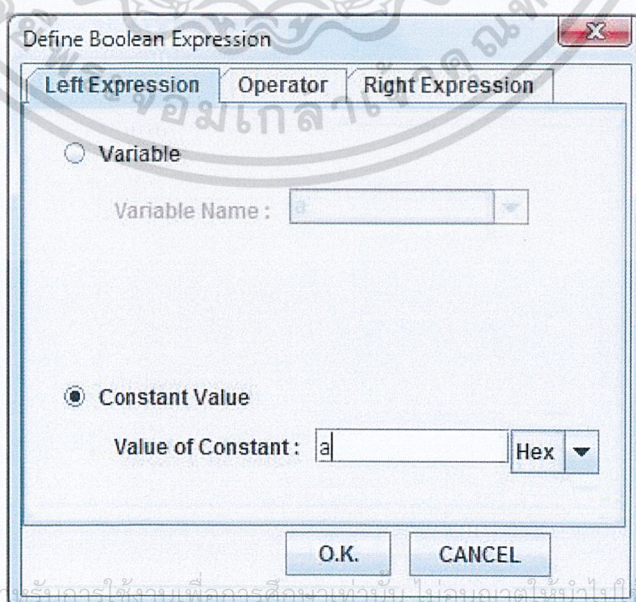
การกำหนดค่าของโมเดล While ทำได้เช่นเดียวกับ โมเดล Condition



รูปที่ ข.14 Pop up Menu ที่ทำการกำหนดค่าของโมเดล While

#### 4.6 โมเดล EndDoWhile

การกำหนดค่าของโมเดล EndDoWhile นั้นก็สามารถทำได้เช่นเดียวกับ โมเดล Condition



รูปที่ ข.15 Pop up Menu ที่ทำการกำหนดค่าของโมเดล EndDoWhile

#### 4.7 โมเดล For

การกำหนดค่าสำหรับ โมเดล For นั้นจะต้องกำหนดค่าของชื่อตัวแปร ค่าเริ่มต้น ค่าสิ้นสุด และเลือกลักษณะของลูปคือสามารถเลือกได้ทั้งแบบนับขึ้นหรือนับลง ดังรูปที่ ข.16

รูปที่ ข.16 Pop up Menu ที่ทำการกำหนดค่าของโมเดล For

#### 4.10 โมเดล Buzzer

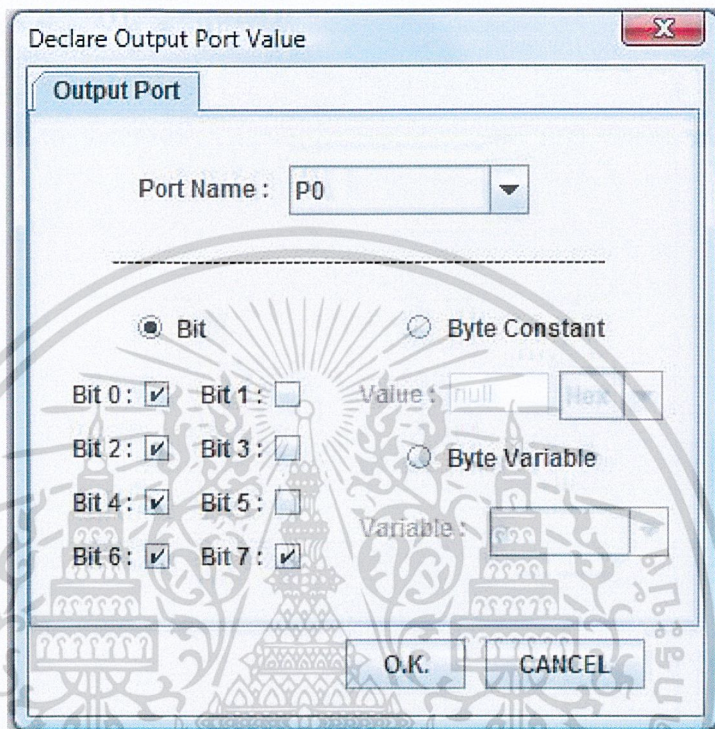
การกำหนดค่าของโมเดล Buzzer นั้นจะต้องระบุเวลาของ buzzer ซึ่งสามารถเลือกเป็นหน่วยวินาทีหรือมิลลิวินาที ดังรูปที่ ข.17

รูปที่ ข.17 Pop up Menu ที่ทำการกำหนดค่าของโมเดล Buzzer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.11 โมเดล LED

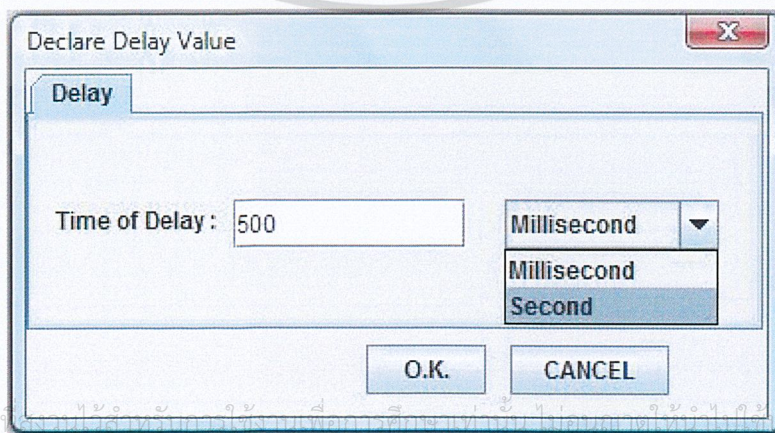
การกำหนดค่าของโมเดล LED นั้นจะต้องระบุพอร์ตที่ติดต่อกับไมโครคอนโทรลเลอร์ เมื่อทำการเลือกแล้ว ผู้ใช้สามารถกำหนดระหว่าง Bit กับ Byte Constant ได้อย่างใดอย่างหนึ่งเท่านั้น ดังรูปที่ ข.18



รูปที่ ข.18 Pop up Menu ที่ทำการกำหนดค่าของโมเดล LED

#### 4.11 โมเดล Delay

การกำหนดค่าของโมเดล Delay นั้นจะต้องระบุเวลาของดีเลย์ โดยสามารถเลือกหน่วยเป็น มิลลิวินาทีหรือนาทีได้ ดังรูปที่ ข.19



เอกสารนี้เป็นเอกสารที่...  
รูปที่ ข.19 Pop up Menu ที่ทำการกำหนดค่าของโมเดล Delay

#### 4.12 โมเดล Number

การกำหนดค่าของโมเดล Number นั้นจะต้องระบุตัวเลขตั้งแต่ 0-9 จำนวนไม่เกิน

8 หลัก

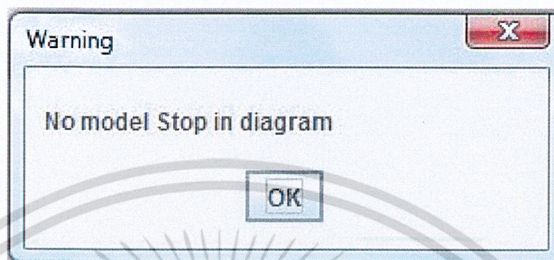
รูปที่ ข.20 Pop up Menu ที่ทำการกำหนดค่าของ โมเดล Number

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5. การ generate code

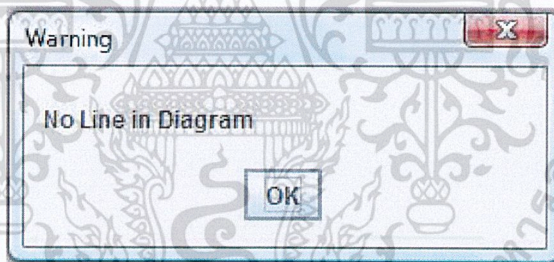
ก่อนที่จะ generate code ได้นั้นผู้ใช้ต้องทำการตรวจสอบ model และตรวจสอบ syntax ก่อนหากการตรวจสอบอย่างใดอย่างหนึ่งผิดพลาดโปรแกรมนี้ไม่สามารถ generate code ให้ได้และมีข้อความผิดพลาดแสดงเพื่อให้ผู้ใช้ทราบถึงข้อผิดพลาดนั้นๆ ยกตัวอย่างเช่น

- กรณีที่โฟลว์ชาร์ตนั้นไม่มีโมเดล Stop จะแสดง Error Dialog ดังรูปที่ ข.21



รูปที่ ข.21 Error Dialog กรณีที่ไม่มีโมเดล Stop ในโฟลว์ชาร์ต

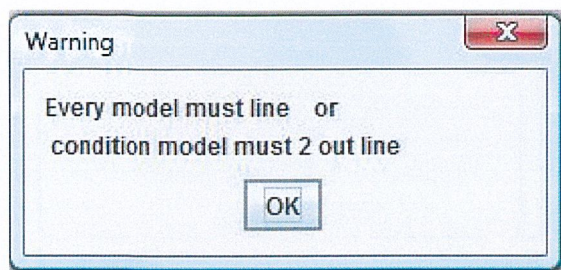
- กรณีที่ไม่มีการลากเส้นเชื่อมใดๆ ระหว่างโมเดล จะแสดง Error dialog ดังรูปที่ ข.22



รูปที่ ข.22 Error Dialog กรณีที่ไม่มีการลากเส้นเชื่อมใดๆ ระหว่างโมเดล

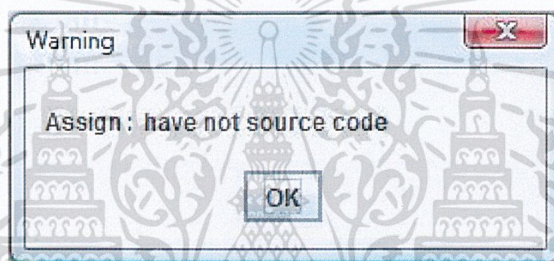
- กรณีที่มีการลากเส้นเชื่อมระหว่างโมเดลไม่ครบทุกโมเดลและโมเดล Condition มีเส้นออกไม่ครบ 2 เส้น จะแสดง Error Dialog ดังรูปที่ ข.23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



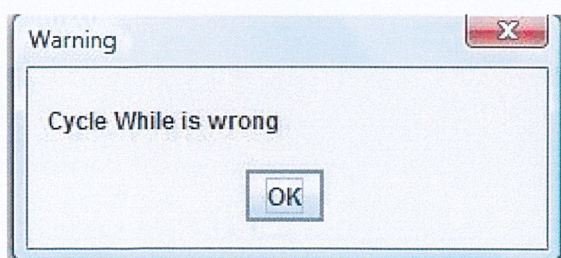
**รูปที่ ข.23** กรณีที่มีการลากเส้นเชื่อมระหว่าง โมเดล ไม่ครบทุกโมเดลและ โมเดล Condition มีเส้น ออกไม่ครบ 2 เส้น

- กรณีที่โมเดลนั้นจำเป็นจะต้องมีการระบุค่าและหากผู้ใช้ยังไม่ได้มีการระบุให้กับ โมเดล จะแสดง Error Dialog เพื่อให้ผู้ใช้ได้กลับไปแก้ไขเพิ่มเติมโค้ดในตัวโมเดลนั้น ดังรูปที่ ข.24



**รูปที่ ข.24** Error Dialog กรณีที่โมเดล Input ไม่มีโค้ด

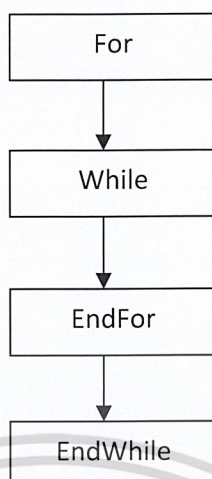
- กรณีที่โมเดลนั้นจำเป็นจะต้องมีการปิดรูป ได้แก่ โมเดล Condition, โมเดล While, โมเดล For, โมเดล EndDoWhile โมเดลเหล่านี้จะต้องมี 2 โมเดลเสมอ เช่น หากมีโมเดล While จำเป็น จะต้องต้องมีโมเดล EndWhile หากมีโมเดลไม่ครบทั้ง 2 โมเดลนี้โปรแกรมจะแสดง Error Dialog เพื่อให้ผู้ใช้ได้กลับไปแก้ไขไฟล์ซอร์สให้ถูกต้อง



**รูปที่ ข.25** Error Dialog กรณีที่โมเดล While ไม่สมบูรณ์

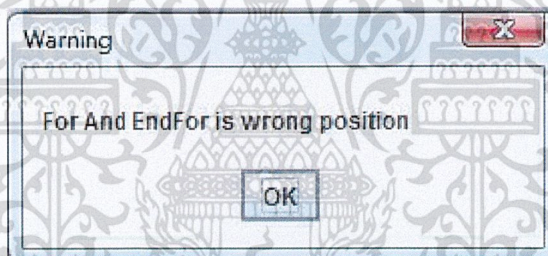
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- กรณีที่โมเดลมีวางตำแหน่งไม่ถูกต้อง เช่น หากมีการจำลองส่วนหนึ่งของโพลีชาร์ตดังนี้



รูปที่ ข.26 โพลีชาร์ตที่มีการวางโมเดลไม่ถูกตำแหน่ง

จากรูปที่ ข.26 จะเห็นว่าโมเดล For และ EndFor ที่มีการวางไม่ถูกตำแหน่ง ในกรณีนี้โปรแกรมจะแสดง Error Dialog ดังรูปที่ ข.27



รูปที่ ข.27 Error Dialog กรณีที่1 โมเดล For วางผิดตำแหน่ง

การตรวจสอบ model สามารถกดที่เมนู Tools >> Check Model และการตรวจสอบ syntax สามารถกดที่เมนู Tools >> Check Syntax ได้เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



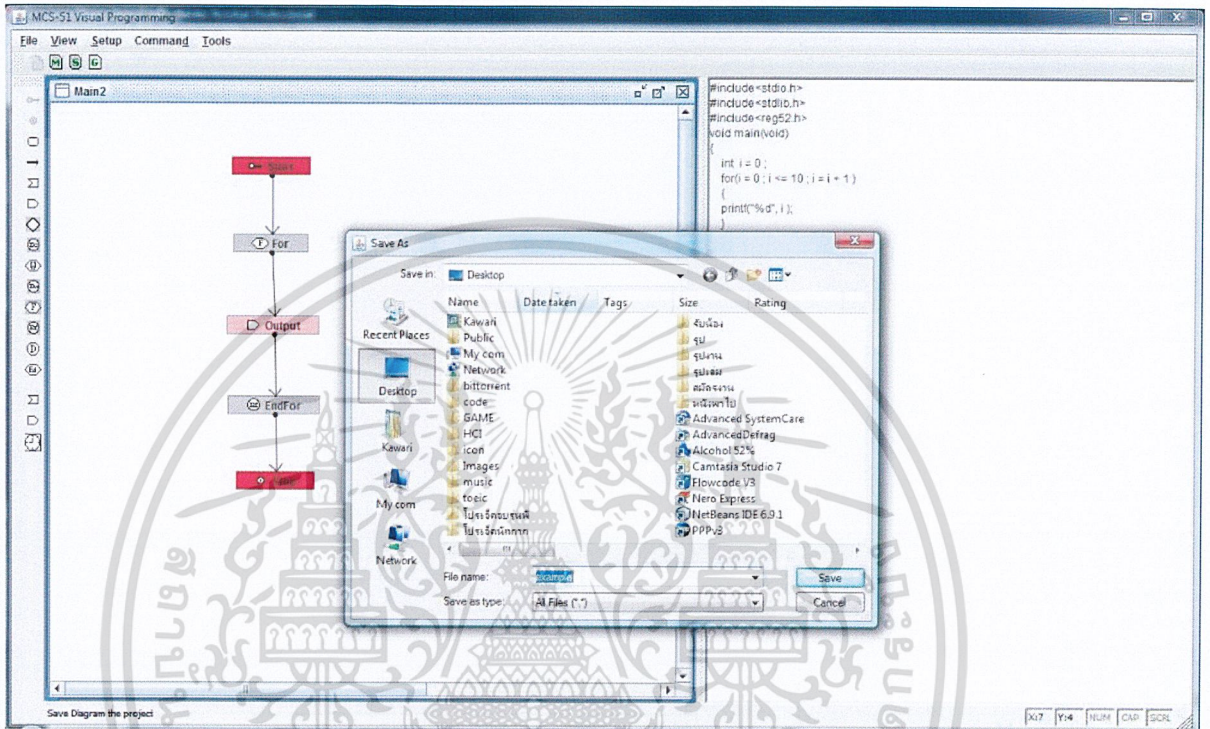
รูปที่ ข.28 หน้าจอหลังจากการกดเมนู generate code

จากรูปที่ ข.28 จะเห็นได้ว่าการกดเมนู generate code แล้วจะได้ code ออกมาทางด้านขวามือของหน้าจอโปรแกรม Visual Programming

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 6. การบันทึกผังงาน

ในการบันทึกผังงานนี้ผู้ใช้สามารถกดที่เมนู file >> Save Diagram หลังจากนั้นทำการเลือก path ที่ต้องการนำไปเก็บ เมื่อผู้ใช้ต้องการเปิดผังงานที่บันทึกไว้สามารถกดที่เมนู file>> Open Diagram โดยในการบันทึกผังงานนั้นจะมีการกำหนดนามสกุลให้อย่างอัตโนมัติ เป็น .mcs



รูปที่ ข.29 หน้าจอเมื่อมีการกดเมนู file >> Save Diagram

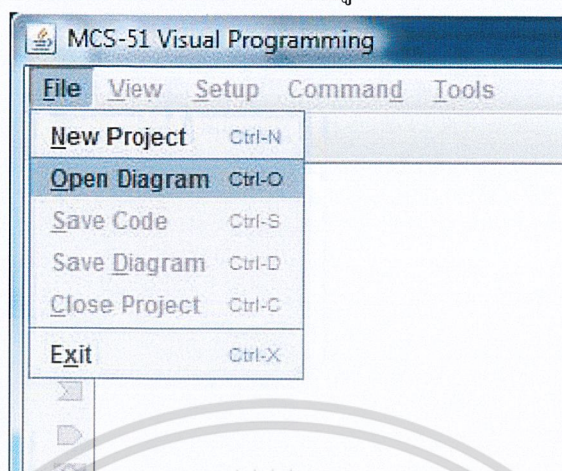
## 7. การบันทึกโค้ด

ในการบันทึกผังงานนี้ผู้ใช้สามารถกดที่เมนู file >> Save Code หลังจากนั้นทำการเลือก path ที่ต้องการนำไปเก็บ ส่วนของหน้าจอนั้นก็จะเป็นเช่นเดียวกับการบันทึกผังงาน แต่นามสกุลจะแตกต่างกันไป คือในการบันทึกโค้ดนั้นมีการกำหนดนามสกุลให้เป็น .c อย่างอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 8. การเปิดผังงานมาใช้ใหม่

การเปิดผังงานมาใช้ใหม่นี้ทำได้โดยเลือกที่เมนู



รูปที่ ข.30 หน้าของ โปรแกรมเมื่อมีการเรียกใช้ผังงานมาใช้ใหม่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ก

### ตัวอย่างขั้นตอนการสร้างโปรแกรม

ในการสร้างโปรแกรมเพื่อใช้โปรแกรม Visual Programming สร้างโค้ดภาษาซีออกมาอัตโนมัติ นั้น ต้องทำการสร้างโปรแกรมให้ถูกวิธีเสียก่อน โดยผู้ใช้ต้องสร้าง syntax ของภาษาซีที่ถูกต้องและระบุค่าในโมเดลทุกตัว ในส่วนนี้จะทำการยกตัวอย่างโปรแกรมเพื่อให้อันดับทดลองทำงานตามตัวอย่างที่กำหนดไว้ ซึ่งจะประกอบไปด้วยขั้นตอนในการสร้างโปรแกรมซึ่งครอบคลุมไปถึงการระบุค่าในโมเดล โดยมีทั้งหมด 2 ตัวอย่าง ได้แก่

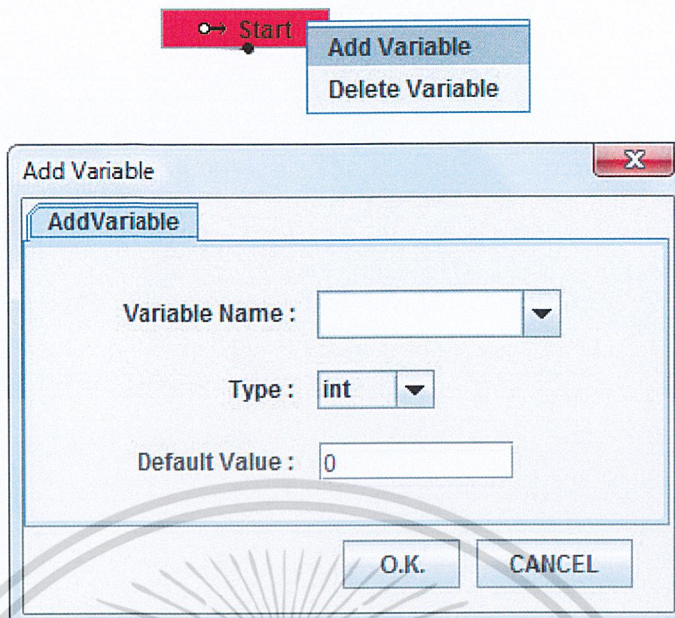
#### ตัวอย่างที่ 1

ในตัวอย่างนี้กำหนดให้อันดับทดลองทำงาน โดยมีเลข 1 แสดงที่หน้าจอแสดงผลในหลักที่ 1 หลอดไฟหลอดที่ 1 ติด และมีเสียงดัง 1 ครั้งนาน 500 มิลลิวินาที หลังจากนั้นหน่วงเวลา 1.5 วินาที แล้วหน้าจอแสดงตัวเลขแสดงเลข 2 ในหลักที่ 1 หลอดไฟดวงที่ 1 และ 2 ติด และมีเสียงดัง 2 ครั้ง ในเวลาเดียวกัน

#### ขั้นตอนในการสร้างโปรแกรมและระบุค่าในโมเดล

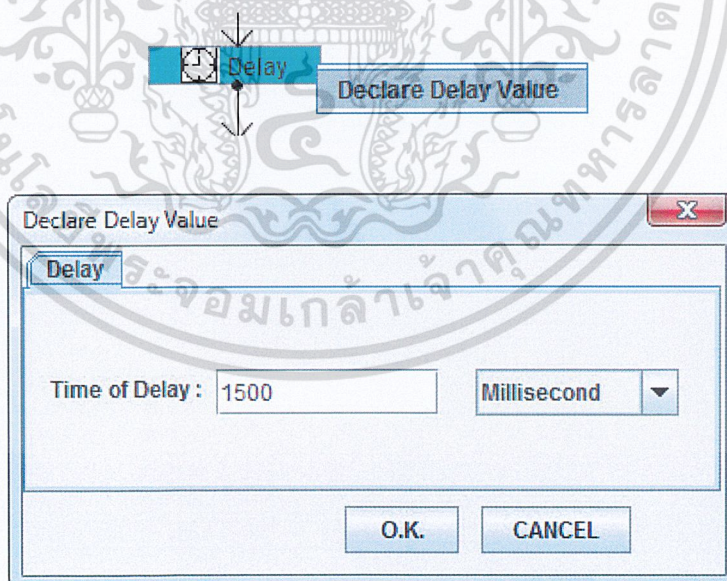
1. คลิกโมเดล Start จาก Model Bar หรือ Tool Bar >> Command จากนั้นคลิกขวาที่โมเดล Start แล้วเลือก Add Variable เพื่อทำการใส่ค่าตัวแปร โดยมีแอททริบิวต์ดังนี้คือ ชื่อตัวแปร ประเภทตัวแปร และค่าของตัวแปร

ในที่นี้ชื่อตัวแปรไม่ต้องใส่ค่าเนื่องจากตัวอย่างนี้ไม่จำเป็นต้องมีการส่งผ่านค่าตัวแปรไปยังโมเดลอื่นๆ ประเภทตัวแปรคือประเภท int และค่าของตัวแปรคือ 0 แล้วกด OK



รูปที่ ค.1 Pop up Menu ของโมเดล Start ตัวอย่างที่ 1

2. คลิกโมเดล Delay แล้วคลิกขวาที่โมเดล จากนั้นเลือกการประกาศค่าให้ Delay โดยโมเดลนี้มีแอททริบิวต์เวลาของDelay และหน่วยของเวลาซึ่งมีมิลลิวินาทีและวินาที ในที่นี้กำหนดให้เวลาของDelay คือ 1500 มิลลิวินาที แล้วกด OK เพื่อหน่วยเวลา 1500 มิลลิวินาทีหรือ 1 นาทีครึ่ง ก่อนที่บอร์ดทดลองจะแสดงตัวเลข หลอดไฟ และเสียง

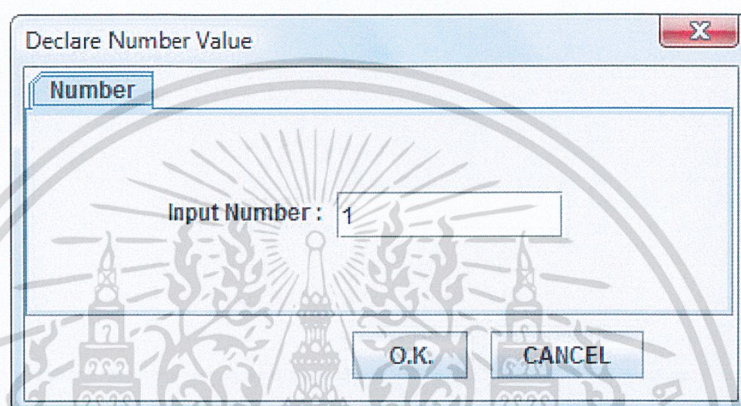
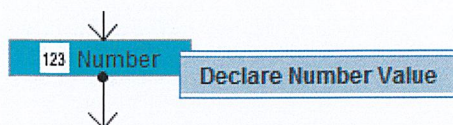


รูปที่ ค.2 Pop up Menu ของโมเดล Delay ตัวอย่างที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. คลิกโมเดล Number แล้วคลิกขวาที่โมเดล จากนั้นเลือกการประกาศค่าให้ Number โดยในโมเดลนี้มีแอททริบิวต์ค่าของตัวเลข โดยสามารถใส่ได้เต็มที่ 8 หลัก หากใส่เกินหรือไม่ใส่ตัวเลขอะไรเลย ระบบจะทำการแจ้งเตือน

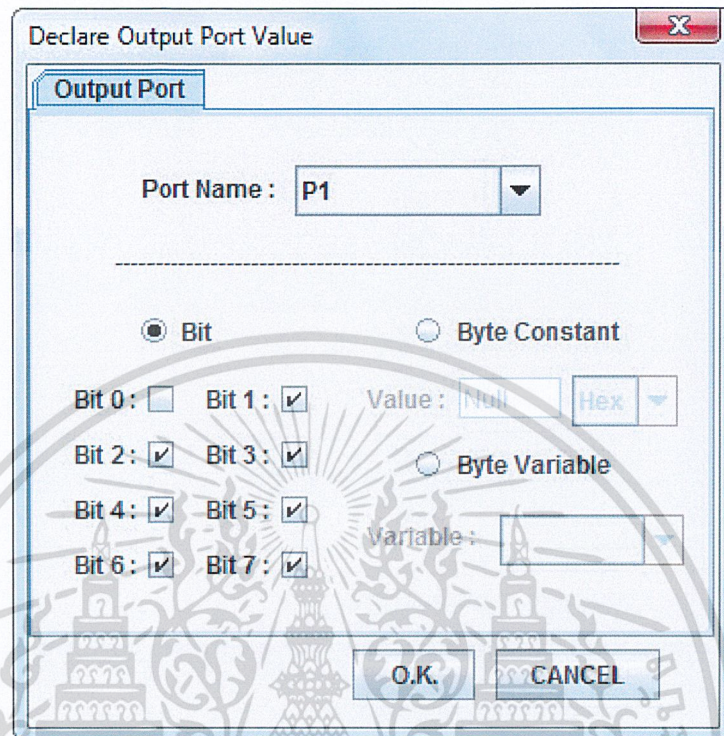
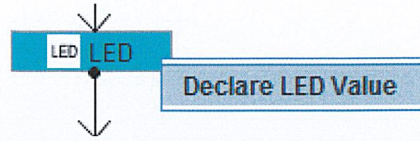
ในที่นี้กำหนดให้ค่า Number คือ 1 เพราะจากโจทย์ตัวอย่าง ต้องการให้บอร์ดทดลองแสดงเลข 1



รูปที่ ค.3 Pop up Menu ของโมเดล Number ตัวอย่างที่ 1

4. คลิกโมเดล LED แล้วคลิกขวาที่โมเดล จากนั้นเลือกการประกาศค่าให้ LED โดยมีแอททริบิวต์ดังนี้คือ ชื่อพอร์ตที่ทำการเชื่อมต่อกับบอร์ดทดลอง ประเภทของค่า LED โดยมีให้เลือกระหว่างประเภท Bit กับ Byte โดยในประเภท Bit นั้นหากทำการคลิกเลือกแล้วคือไฟดับ หากช่องไหนปล่อยว่างไว้คือไฟดับ

ในที่นี้คือบอร์ดทดลองเชื่อมต่อกับหลอดไฟ LED ที่พอร์ต 1 จึงเลือก P1 และเลือกประเภทของ LED แบบ Bit เพื่อง่ายในการทำความเข้าใจ คลิกที่ Bit 1-7 เพื่อให้หลอดไฟดวงที่ 2-8 ดับ ส่วน Bit 0 ปล่อยว่างไว้เพราะต้องการให้หลอดไฟดวงแรกสว่าง

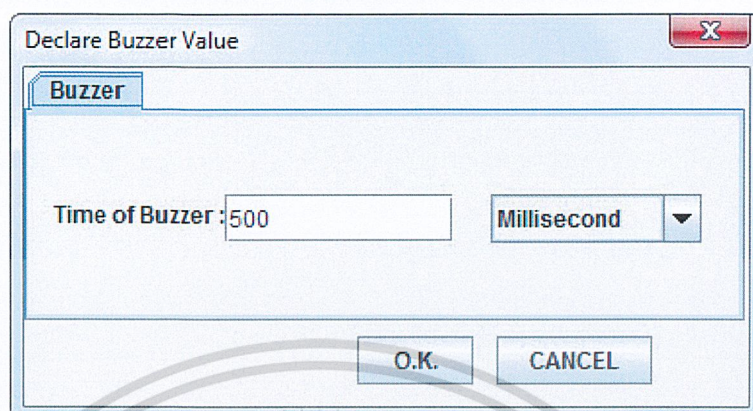
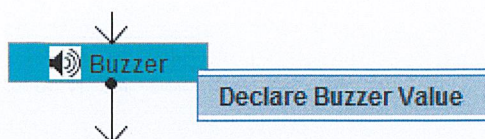


รูปที่ ค.4 Pop up Menu ของโมเดล LED ตัวอย่างที่ 1

5. คลิก โมเดล Buzzer แล้วคลิกขวาที่โมเดล จากนั้นเลือกการประกาศค่าให้ Buzzer โดยมี แอททริบิวต์เวลาการหน่วงเวลาของ Buzzer และหน่วยของเวลาซึ่งมีมิลลิวินาทีและวินาที

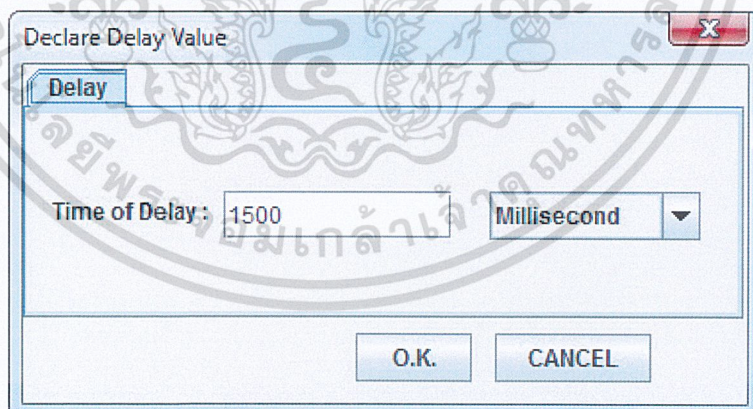
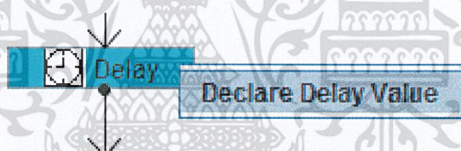
ในที่นี้เวลาการหน่วงเวลาของ Buzzer กำหนดไว้คือ 500 มิลลิวินาที จากการใส่ค่าของ โมเดลนี้ Buzzer จะมีเสียงดังนาน 500 มิลลิวินาทีหรือครั้งนาที่นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.5 Pop up Menu ของโมเดล Buzzer ตัวอย่างที่ 1

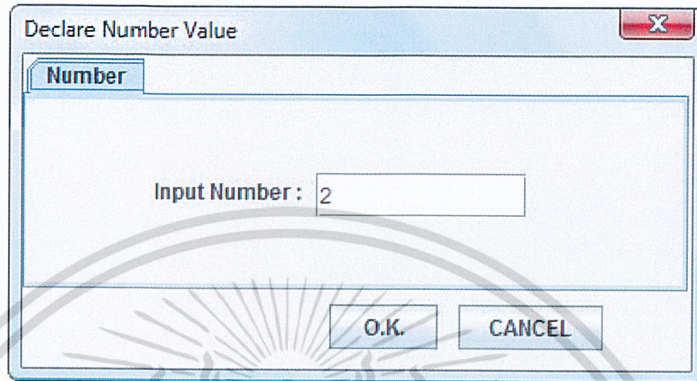
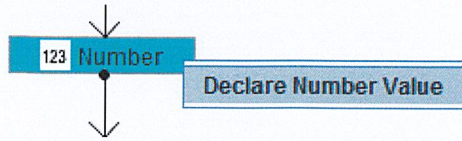
6. คลิกโมเดล Delay แล้วคลิกขวาที่โมเดล จากนั้นกำหนดให้เวลาของ Delay คือ 1500 มิลลิวินาที แล้วกด OK เพื่อหน่วงเวลาก่อนที่บอร์ดทดลองจะแสดงตัวเลข หลอดไฟและเสียงอีกครั้ง



รูปที่ ค.6 Pop up Menu ของโมเดล Delay ที่ 2 ตัวอย่างที่ 1

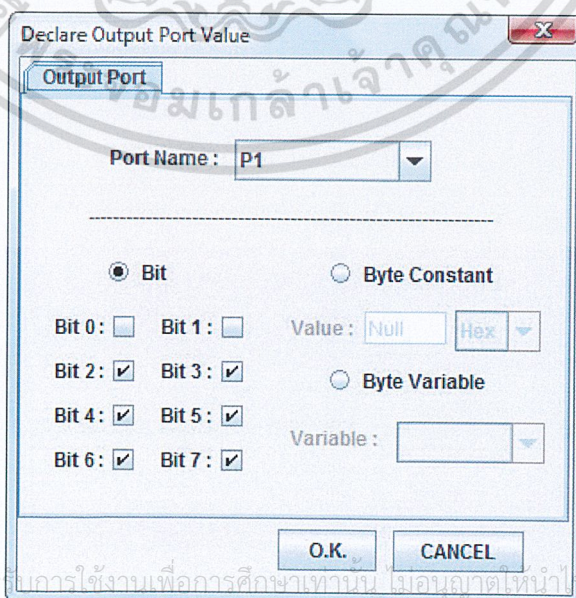
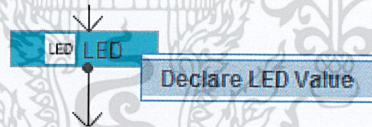
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. คลิกโมเดล Number แล้วคลิกขวาที่โมเดล จากนั้นเลือกการประกาศค่าให้ Number โดยในที่นี้กำหนดให้ค่า Number คือ 2 เพราะในตัวอย่างนี้ ต้องการให้บอร์ดทดลองแสดงเลข 2 นั่นเอง



รูปที่ ค.7 Pop up Menu ของโมเดล Number ที่ 2 ตัวอย่างที่ 1

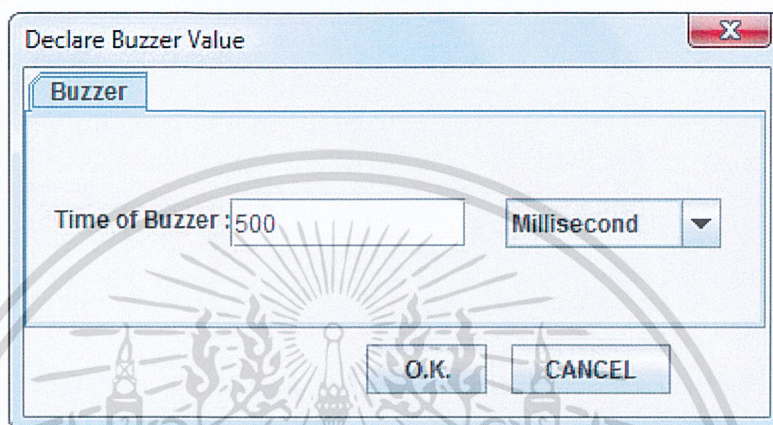
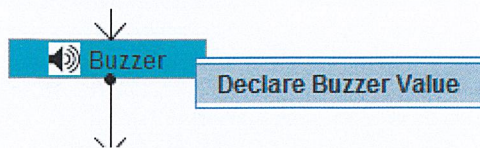
8. คลิกโมเดล LED แล้วคลิกขวาที่โมเดล จากนั้นเลือกการประกาศค่าให้ LED ในที่นี้บอร์ดทดลองเชื่อมต่อกับพอร์ต 1 จึงเลือก P1 และเลือกประเภทของ LED แบบ Bit คลิกที่ Bit 2-7 เพื่อให้หลอดไฟดวงที่ 3-8 ดับ ส่วน Bit 0 และ Bit 1 ปล่อยให้ว่างไว้เพราะต้องการให้หลอดไฟดวงแรกและดวงที่ 2 สว่าง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ข้อมูลใดๆ ทั้งสิ้น

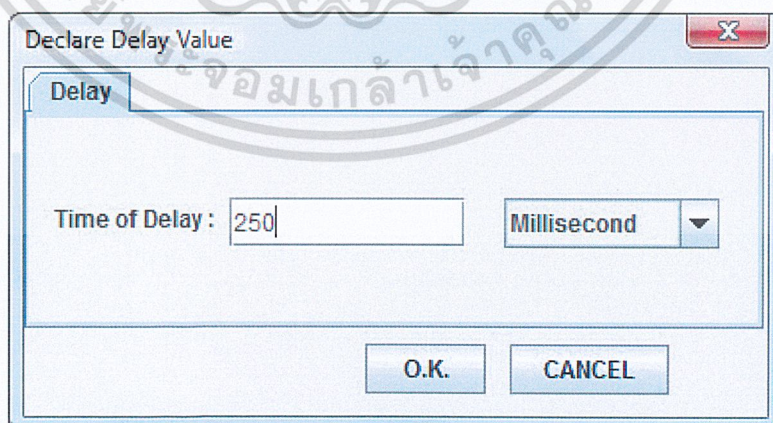
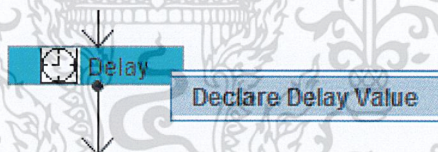
รูปที่ ค.8 Pop up Menu ของโมเดล LED ที่ 2 ตัวอย่างที่ 1 ครั้งที่มีการนำไปใช้

9. คลิกโมเดล Buzzer แล้วคลิกขวาที่โมเดล จากนั้นเลือกการประกาศค่าให้ Buzzer โดยเวลาการหน่วงเวลาของ Buzzer กำหนดไว้คือ 500 มิลลิวินาที นั่นคือจะมีเสียงดังเป็นเวลาครั้งวินาทีนั่นเอง



รูปที่ ค.9 Pop up Menu ของโมเดล Buzzer ที่ 2 ตัวอย่างที่ 1

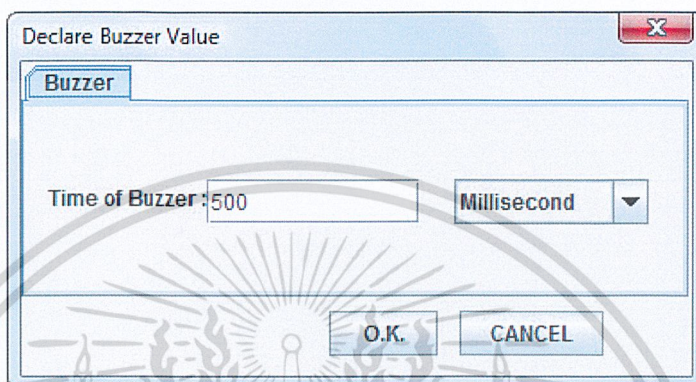
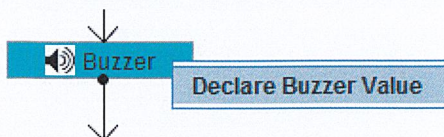
10. คลิกโมเดล Delay แล้วคลิกขวาที่โมเดล จากนั้นกำหนดให้เวลาของ Delay คือ 250 มิลลิวินาที แล้วกด OK เพื่อให้หน่วงเวลาหลังจากมีเสียงดังจากข้อ 11.



รูปที่ ค.10 Pop up Menu ของโมเดล Delay ที่ 3 ตัวอย่างที่ 1

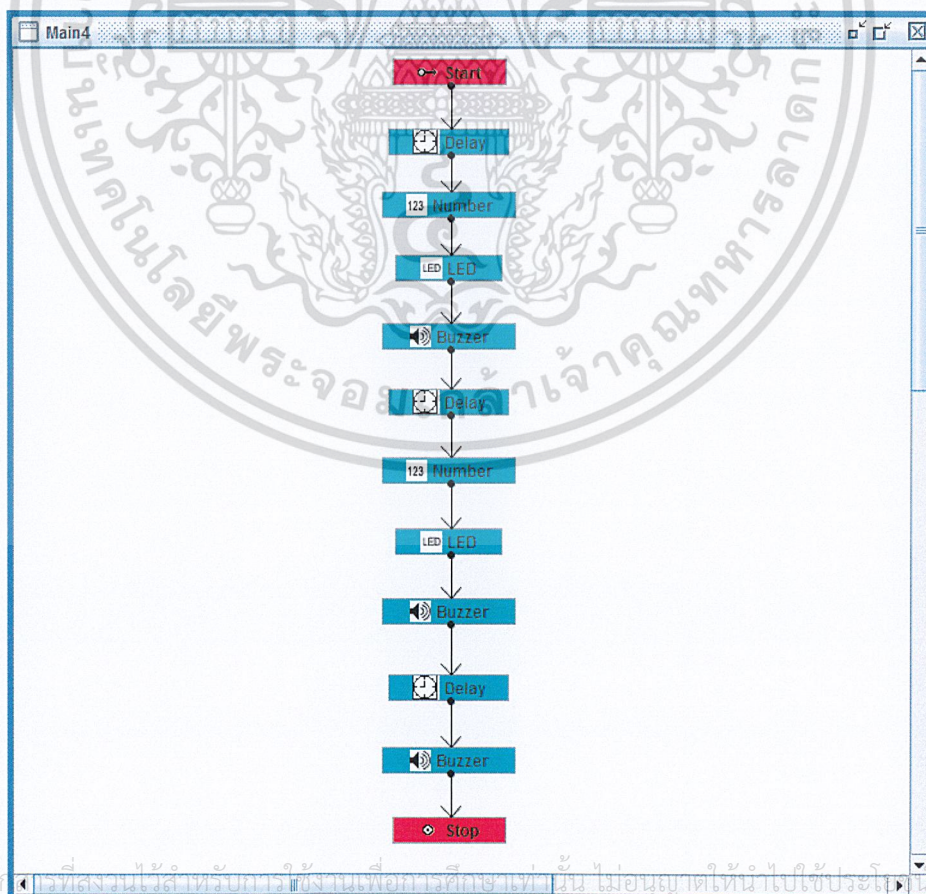
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. คลิกโมเดล Buzzer แล้วคลิกขวาที่โมเดล จากนั้นเลือกการประกาศค่าให้ Buzzer โดยเวลาการหน่วงเวลาของ Buzzer กำหนดไว้คือ 500 มิลลิวินาที จากการทดสอบจะเห็นได้ว่า หลังจากมีเสียงดังนานครึ่งวินาทีแล้ว จะมีการหน่วงเวลา และมีเสียงดังขึ้นอีกนั่นเอง



รูปที่ ค.11 Pop up Menu ของโมเดล Delay ที่ 3 ตัวอย่างที่ 1

12. แผนผังโฟลว์ชาร์ตเมื่อมีโมเดล Start แล้วต้องมีโมเดล Stop ปิดเสมอ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ต่อรูปที่ ค.12 แผนผังโฟลว์ชาร์ตตัวอย่างที่ 1

13. เมื่อสร้างโฟลว์ชาร์ตพร้อมระบุค่าในโมเดลเรียบร้อยแล้ว ให้ทำการกด Check Model, Check Syntax และ Generate Code ตามลำดับ หากโฟลว์ชาร์ตที่สร้างขึ้นมีข้อผิดพลาดจะมีข้อความแจ้งเตือนขึ้น ผู้ใช้สามารถย้อนกลับไปแก้ไขให้ถูก แล้วดำเนินการตามขั้นตอนเดิม

```
#include<stdio.h>
#include<stdlib.h>
#include<reg52.h>
#include<myfunction.h>
#include<shownum.h>
void main(void)
{
    dmsec(1500);

    dis_num(1);

    P1 = 0xFE;

    beep(500);

    dmsec(1500);

    dis_num(2);

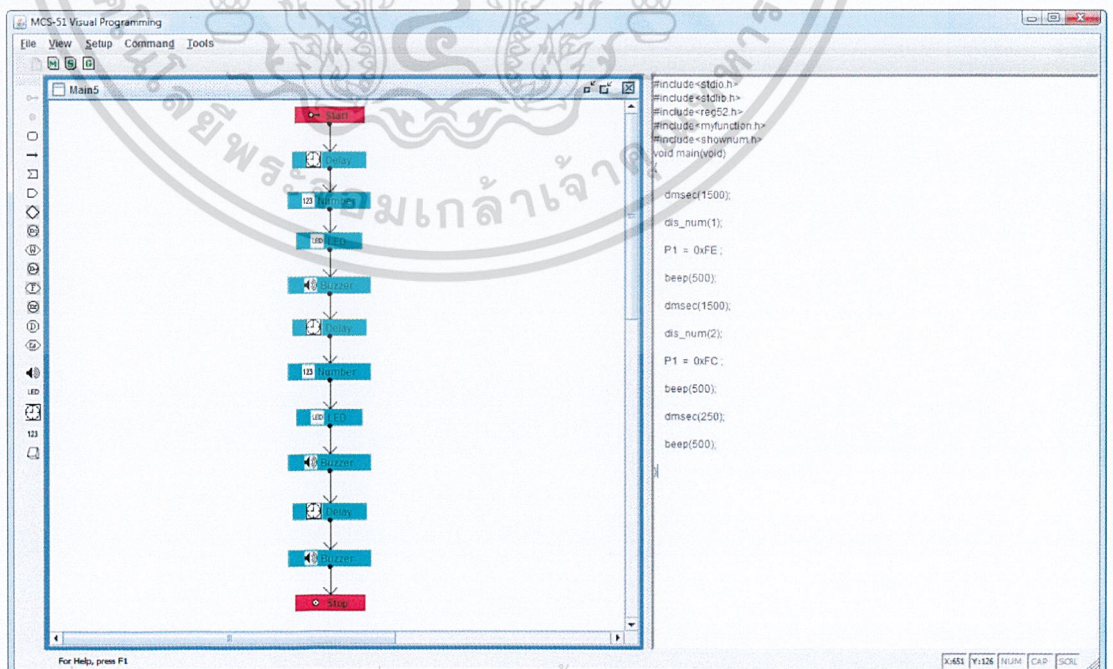
    P1 = 0xFC;

    beep(500);

    dmsec(250);

    beep(500);
}
```

รูปที่ ค.13 โค้ดที่ได้หลังการ generate ของตัวอย่างโปรแกรมที่ 1



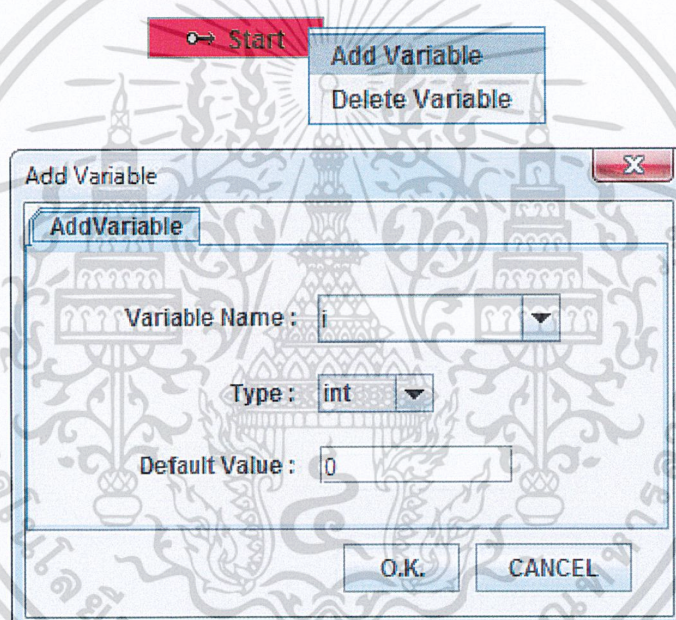
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเพื่อตรวจสอบเท่านั้น ไม่ควรนำไปเผยแพร่โดยไม่ขออนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้  
**รูปที่ ค.14 ภาพรวมของตัวอย่างโปรแกรมที่ 1**  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ตัวอย่างที่ 2

ในตัวอย่างที่ 2 นี้ต้องการให้บอร์ดทดลองทำงานโดยให้รีเลย์และหลอดไฟทั้ง 8 ดวงติดพร้อมกัน หลังจากนั้นทั้งรีเลย์และหลอดไฟดับ 1 วินาที และทำงานซ้ำไปเรื่อยๆ โดยจะมีการนำ For loop และ Condition เข้ามาใช้ในโปรแกรมซึ่งจะมีความซับซ้อนมากกว่าตัวอย่างที่ 1 เพื่อประหยัดพื้นที่ในการเก็บโค้ดและใช้เวลาในการสร้างโฟลว์ชาร์ตในระยะเวลาอันรวดเร็ว

### ขั้นตอนการสร้างโฟลว์ชาร์ตและระบุค่าในโมเดล

1. คลิกโมเดล Start จากนั้นคลิกขวาที่โมเดล Start แล้วเลือก Add Variable เพื่อทำการใส่ค่าตัวแปร ในที่นี้ชื่อตัวแปรกำหนดชื่อคือ i ประเภทตัวแปรคือประเภท int และค่าของตัวแปรคือ 0 แล้วกด OK เพื่อส่งตัวแปร i ไปให้โมเดลอื่นใช้งาน

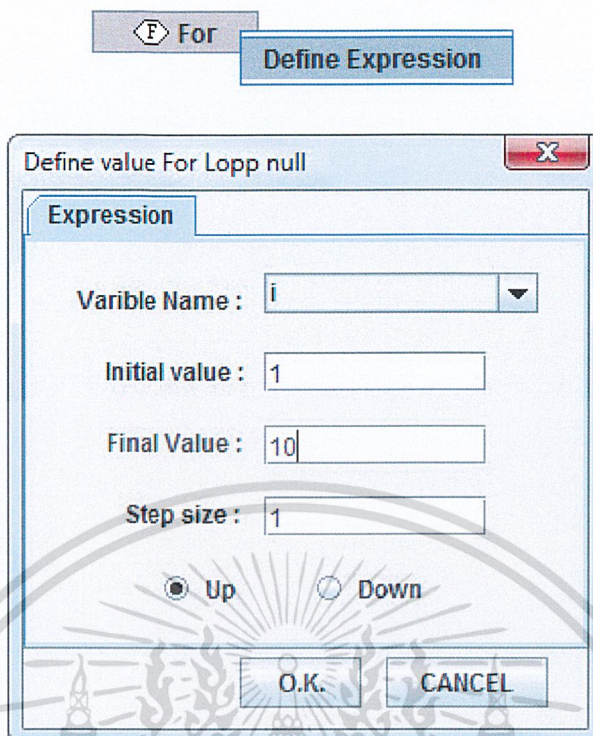


รูปที่ ค.15 Pop up Menu ของโมเดล Start ตัวอย่างที่ 2

2. คลิกโมเดล For Loop จากนั้นคลิกขวาที่โมเดล แล้วเลือกระบุค่าให้กับ For Loop โดยมีเอททริบิวต์ดังนี้คือ ชื่อตัวแปร ค่าเริ่มต้น ค่าสิ้นสุด จำนวนการเพิ่มขึ้นหรือลดลง และประเภทของค่าคือเพิ่มขึ้นหรือลดลง

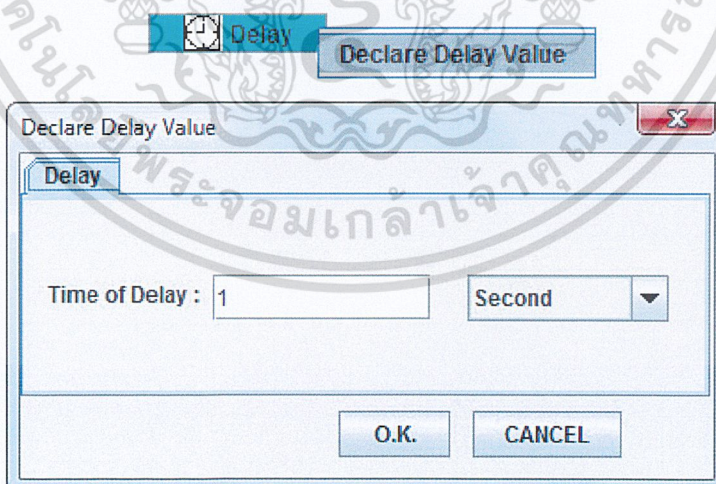
ในที่นี้กำหนดให้ชื่อตัวแปรคือ i ซึ่งเป็นค่าที่ส่งมาจากโมเดล Start ค่าเริ่มต้นคือ 1 ค่าสิ้นสุดคือ 10 เพิ่มขึ้นทีละ 1 ที่กำหนดเช่นนี้เพราะต้องการให้ลูบนี้ทำงานตั้งแต่ 1-10 โดยเพิ่มค่าทีละ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.16 Pop up Menu ของโมเดล For ตัวอย่างที่ 2

3. คลิกโมเดล Delay แล้วคลิกขวาที่โมเดล จากนั้นเลือกการประกาศค่าให้ Delay โดยโมเดลนี้มีแอททริบิวต์เวลาของDelay และหน่วยของเวลาซึ่งมีมิลลิวินาทีและวินาที ในที่นี้กำหนดให้เวลาของDelay คือ 1 วินาที แล้วกด OK เพื่อหน่วยเวลา 1 วินาที

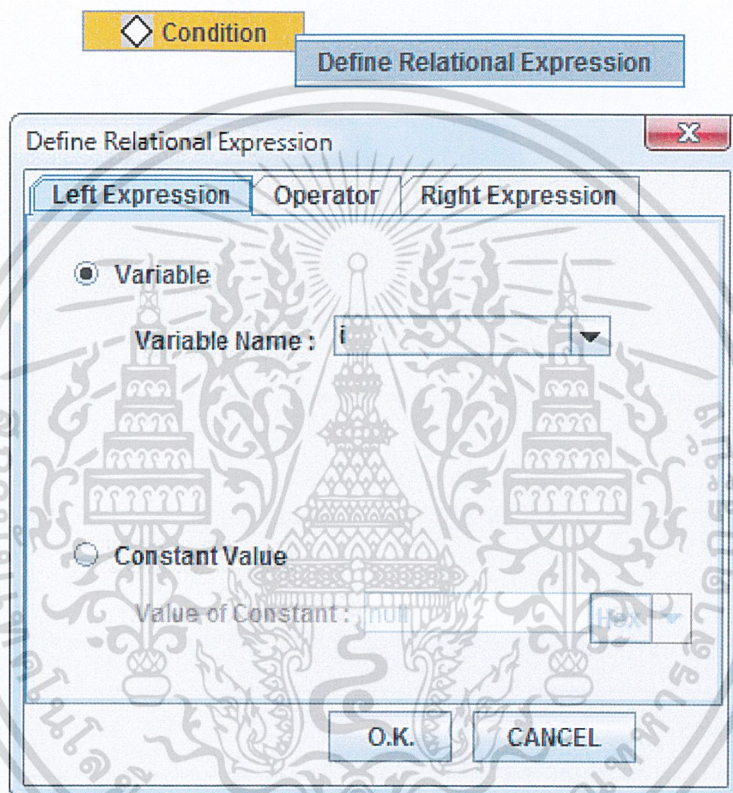


รูปที่ ค.17 Pop up Menu ของโมเดล Delay ตัวอย่างที่ 1

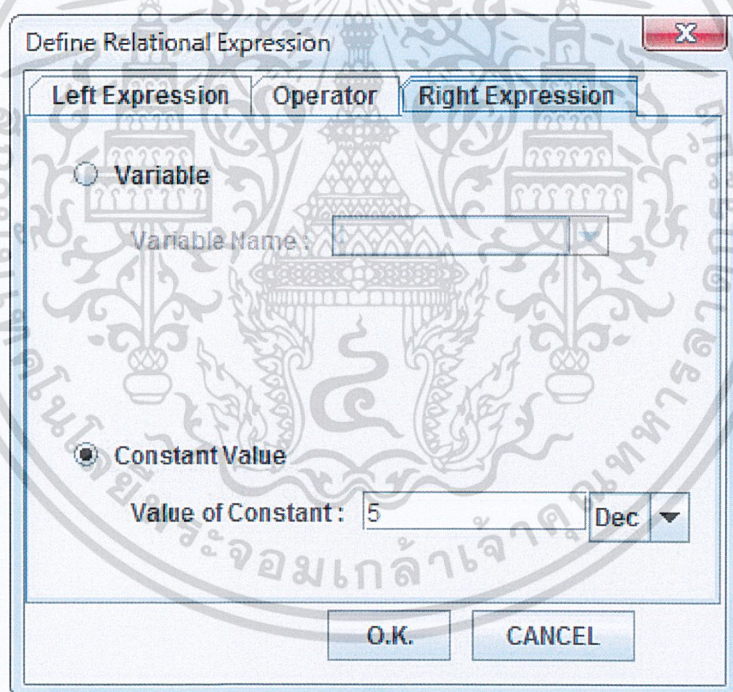
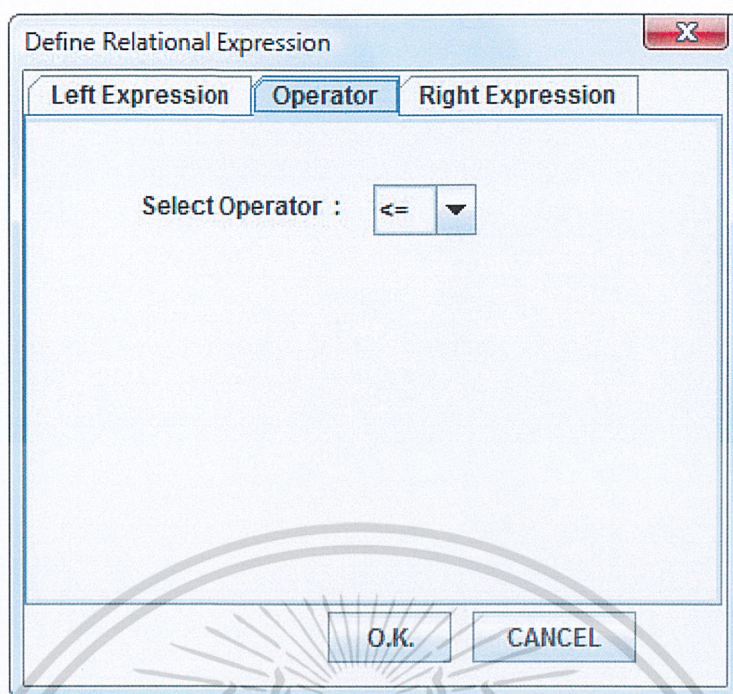
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. คลิกโมเดล Condition แล้วคลิกขวาที่โมเดลเพื่อทำการระบุค่าความสัมพันธ์ โดยจะมี 3 แถบคือ ค่าทางด้านซ้าย ตัวดำเนินการทางคณิตศาสตร์ และค่าทางด้านขวา โดยค่าทางด้านซ้ายและขวามีแอมพริบวต์เหมือนกันคือ ค่าตัวแปรที่ส่งมาจากโมเดล Start และค่าคงที่โดยสามารถกำหนดค่าคงที่ก็ได้

ในที่นี้กำหนดค่าทางด้านซ้ายคือตัวแปรชื่อ i ตัวดำเนินการทางคณิตศาสตร์คือ  $\leq$  และค่าทางด้านขวาคือ ค่าคงที่ กำหนดให้เป็น 5 ในเลขฐานสิบ เพื่อให้ Condition ทำงานภายใต้ For Loop



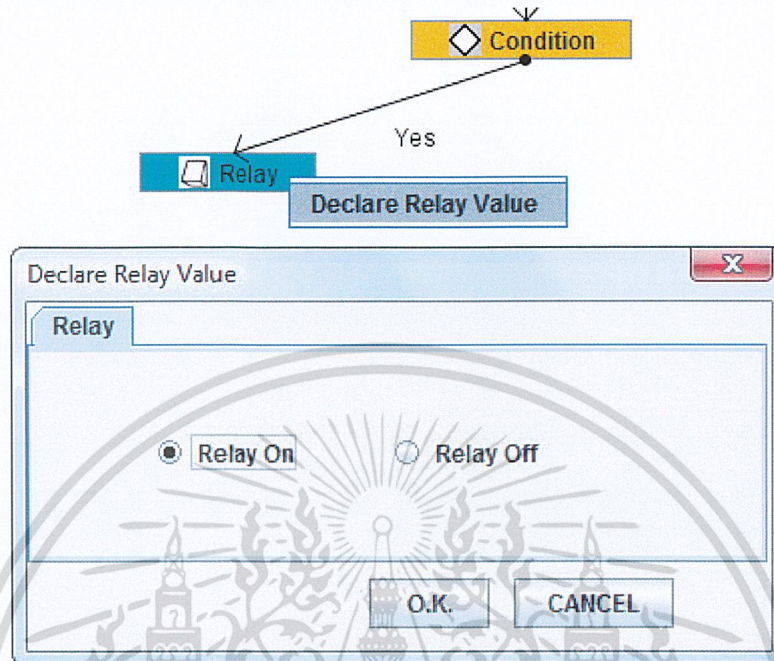
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.18 Pop up Menu ของโมเดล Condition ตัวอย่างที่ 2

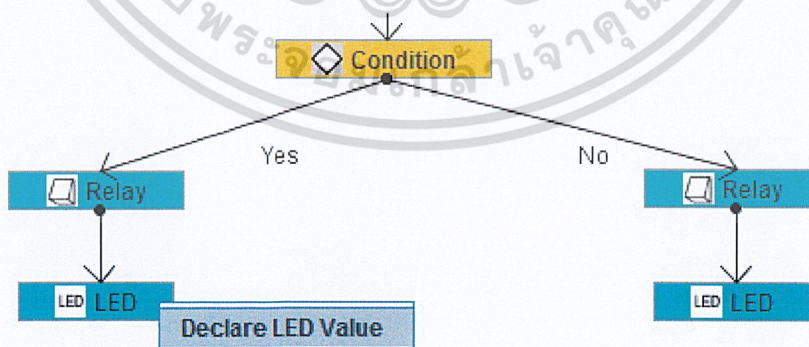
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. หากค่าใน Condition คือ 1-5 ดังข้อ 4 ที่กำหนดขึ้น Relay บนบอร์ดทดลองจะติด ดังนั้นจึงต้องคลิกที่โมเดล Relay จากนั้นคลิกขวาที่โมเดลเพื่อทำการระบุค่าให้ Relay และกำหนดให้ Relay เปิด สังเกตว่าหากลากเส้นออกจากโมเดล Condition ไปทางด้านซ้าย เงื่อนไขจะเป็น Yes เสมอ



รูปที่ ก.19 Pop up Menu ของโมเดล Relay ตัวอย่างที่ 2

6. หากค่าใน Condition คือ 1-5 ดังข้อ 4 หลอด LED จะติดทุกหลอด ดังนั้นต้องคลิกที่โมเดล LED จากนั้นคลิกขวาแล้วทำการระบุค่า โดยกำหนดให้พอร์ตเชื่อมต่อกับหลอดไฟ LED คือพอร์ต 1 และค่าของ LED เป็นแบบ Bit แล้วไม่ต้องคลิก Bit 0 - Bit 7 เพราะต้องการให้หลอดไฟทุกดวงติด จากนั้นคลิก OK



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

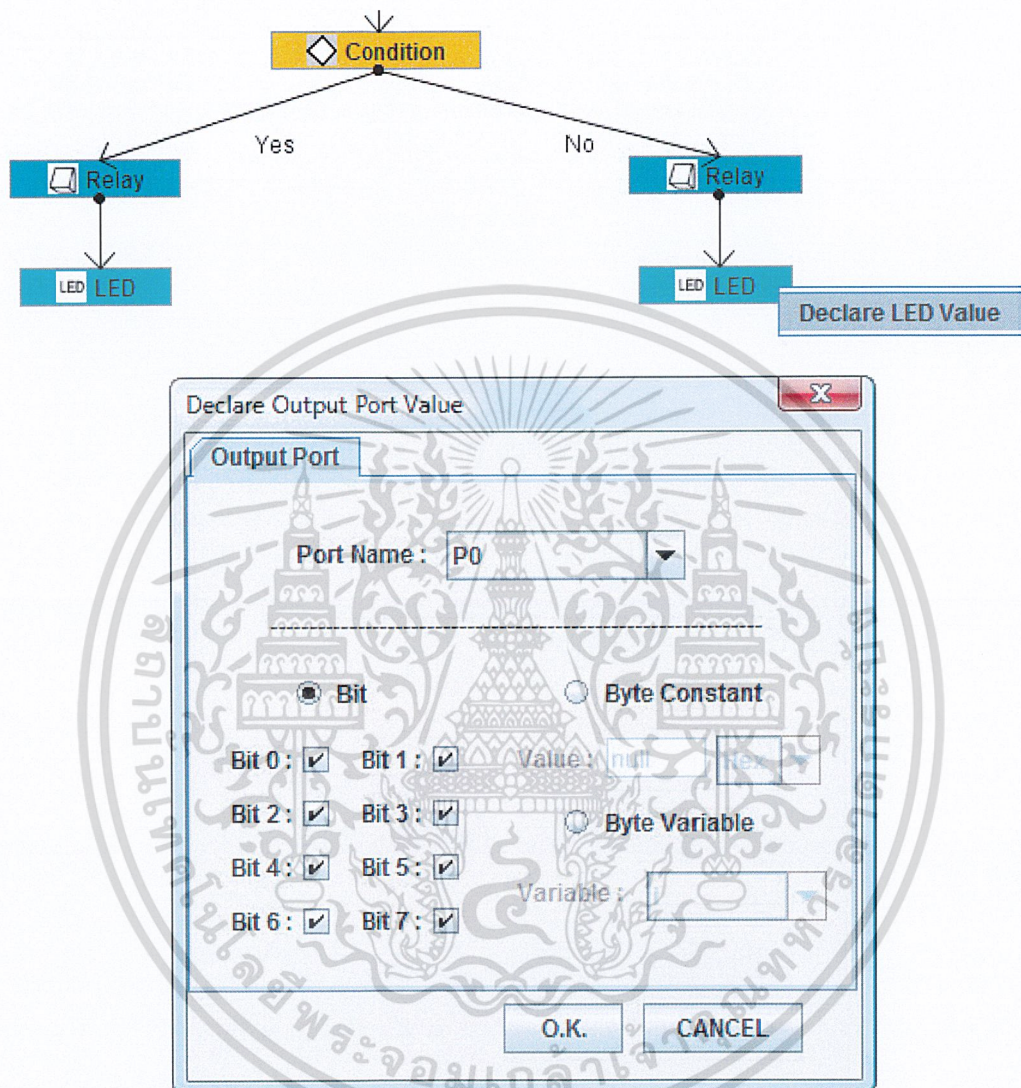
รูปที่ ค.20 Pop up Menu ของ โมเดล LED ตัวอย่างที่ 2

7. หากค่าใน Condition คือ ค่าอื่นที่ไม่ใช่ 1-5 Relay บนบอร์ดทดลองจะดับคั้งนั้นจึงต้องคลิกที่โมเดล Relay จากนั้นคลิกขวาที่โมเดลเพื่อทำการระบุค่าให้Relay และกำหนดให้ Relay ปิด สังเกตได้ว่าหากลากเส้นออกจาก Condition ไปทางด้านขวา เงื่อนไขจะเป็น No เสมอ

รูปที่ ค.21 Pop up Menu ของ โมเดล Relay ที่ 2 ตัวอย่างที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

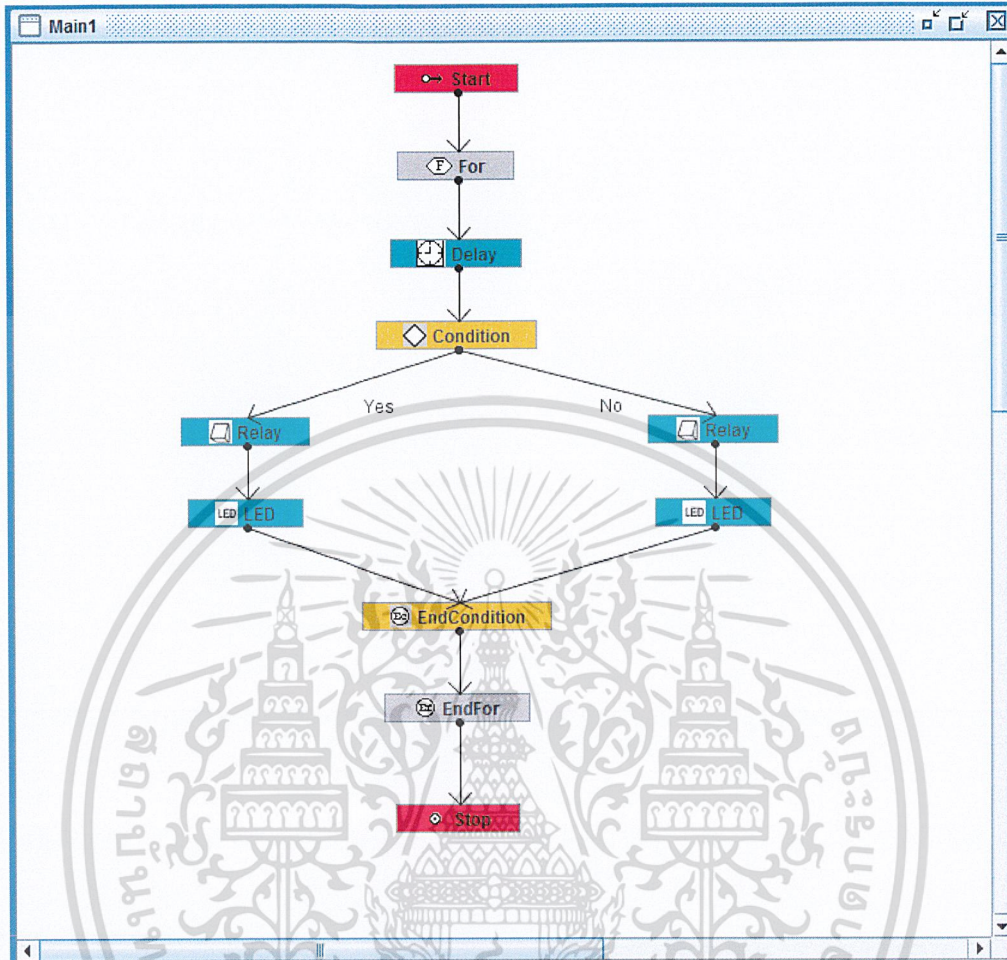
8. หากค่าใน Condition คือค่าอื่นที่ไม่ใช่ 1-5 หลอด LED จะดับทุกหลอด ดังนั้นต้องคลิกที่โมเดล LED จากนั้นคลิกขวาแล้วทำการระบุค่า โดยกำหนดให้พอร์ตเชื่อมต่อกับหลอดไฟ LED คือ พอร์ต 1 และค่าของ LED เป็นแบบ Bit แล้วคลิก Bit 0 – Bit 7 เพราะต้องการให้หลอดไฟทุกดวงติด จากนั้นคลิก OK



รูปที่ ค.22 Pop up Menu ของโมเดล LED ที่ 2 ตัวอย่างที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. แผนผังโฟลว์ชาร์ตเมื่อมีโมเดล Start แล้วต้องมีโมเดล Stop ปิดเสมอ ในโปรแกรมนี้มีการใช้ For Loop และ condition ดังนั้นจึงต้องมี End For Loop และ End Condition ปิดด้วย



รูปที่ ค.23 แผนผังโฟลว์ชาร์ตตัวอย่างที่ 2

10. เมื่อสร้างโฟลว์ชาร์ตพร้อมระบุค่าในโมเดลเรียบร้อยแล้ว ให้ทำการกด Check Model, Check Syntax และ Generate Code ตามลำดับ หากโฟลว์ชาร์ตที่สร้างขึ้นมีข้อผิดพลาดจะมีข้อความแจ้งเตือนขึ้น ผู้ใช้สามารถย้อนกลับไปแก้ไขให้ถูกต้องแล้วดำเนินการตามขั้นตอนเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include<stdio.h>
#include<stdlib.h>
#include<reg52.h>
#include<myfunction.h>
#include<shownum.h>
void main(void)
{
    int i = 0;
    for(i = 1; i <= 10; i = i + 1)
    {

        dmsec(1000);
        if (i <= 5 )
        {

            relay_on();

            P1 = 0x0 ;

        }else
        {

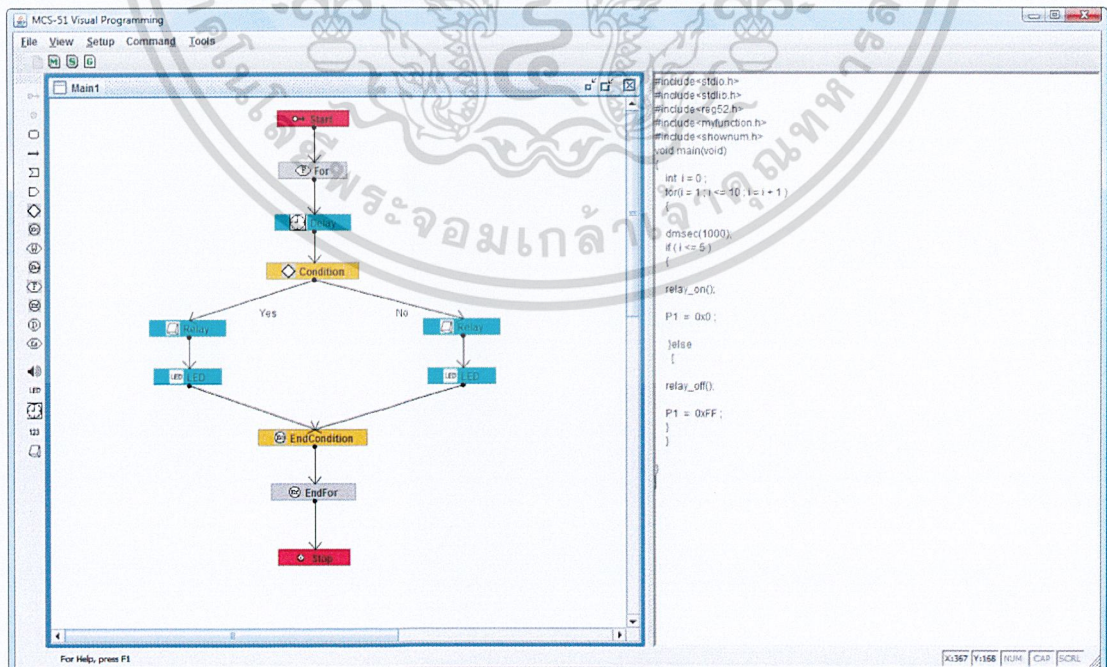
            relay_off();

            P1 = 0xFF ;

        }
    }
}

```

รูปที่ ค.24 โค้ดที่ได้จากการ generate ของตัวอย่างโปรแกรมที่ 2



รูปที่ ค.25 ภาพรวมของตัวอย่างโปรแกรมที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ส่วนตัวของนักศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้