บทที่ 3

วิธีดำเนินการวิจัย

จากการศึกษาทฤษฎี และองค์ประกอบอื่น ๆ ที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์สามารถแบ่งกระบวนการดำเนินการวิจัยได้ดังนี้

3.1 แนวทางในการพัฒนาซอฟต์แวร์ที่ใช้ควบคู่กับฮาร์ดแวร์

3.2 กระบวนการในการพัฒนาซอฟต์แวร์และฮาร์ดแวร์

3.3 การติดตั้งอุปกรณ์ที่ใช้ควบคุมค่าความสว่าง

3.4 การแสดงผลการควบคุมค่าความสว่าง

3.5 โครงสร้างของซอฟต์แวร์

3.1 แนวทางในการพัฒนาซอฟต์แวร์ที่ใช้ควบคู่กับฮาร์ดแวร์

จากปัญหาที่พบในการปรับปรุงประสิทธิภาพการใช้พลังงานไฟฟ้าโดยการปรับปรุงอาคารนั้น ปริมาณการใช้พลังงานไฟฟ้ากลับลดลงเพียงเล็กน้อยเท่านั้น และหากผู้ใช้อาคารไม่มีการปรับพฤติกรรมการใช้พลังงานที่เหมาะสมด้วยแล้ว ก็จะไม่สามารถลดการใช้พลังงานไฟฟ้าลงได้ ดังนั้นตัวซอฟต์แวร์ที่พัฒนาขึ้นนี้ ได้มีการใช้ฮาร์ดแวร์ที่ช่วยควบคุมความส่องสว่างของหลอดไฟให้สามารถปรับค่าความสว่างได้ตามความเหมาะสมของสภาพแวดล้อม ซึ่งซอฟต์แวร์ในปัจจุบันที่ช่วยออกแบบระบบแสงสว่างนั้นยังไม่สามารถควบคุมปรับค่าความสว่างได้ แม้ว่าซอฟต์แวร์สามารถออกแบบและคำนวณได้ แต่ด้วยปัจจัยในการคำนวณที่มีความซับซ้อนเกินไปทำให้ผู้ใช้งานต้องลำบากในการทำความเข้าใจปัจจัยต่าง ๆ ยิ่งไปกว่านั้นผู้ใช้งานส่วนใหญ่ยังขาดความรู้พื้นฐานในการออกแบบแสงสว่าง หรือมีวิธีการออกแบบที่ไม่ถูกต้อง เพราะการออกแบบแสงสว่างมีวิธีการคำนวณที่ซับซ้อน ต้องอาศัยประสบการณ์ความรู้ในเชิงเทคนิค รวมทั้งลักษณะของหลอดไฟมีอยู่หลายชนิด และมีคุณสมบัติที่แตกต่างกันไป ซึ่งเป็นการยากที่ผู้ใช้งานจะสามารถเลือกใช้ได้อย่างถูกต้อง ทำให้เกิดความยุ่งยากและเสียเวลาในการคำนวณหาค่าความสว่าง รวมไปถึงพื้นที่ใช้งานก็ไม่ได้รับแสงสว่างที่พอเหมาะตามประโยชน์ใช้สอยของพื้นที่นั้น ๆ จึงทำให้ผู้ใช้งานไม่สามารถนำไปเปรียบเทียบหรือวิเคราะห์ใช้งานต่อได้ ดังนั้นการพัฒนาซอฟต์แวร์ในงานวิจัยนี้ควรสามารถทำให้ผู้ใช้งานและเจ้าของกิจการเห็นถึงความสำคัญ และสร้างทางเลือกใน

การใช้พลังงานให้มีประสิทธิภาพมากยิ่งขึ้นได้ งานวิทยานิพนธ์นี้จึงได้ทำการศึกษาการจัดวางตำแหน่งหลอดไฟภายในห้องและเพิ่มประสิทธิภาพในการควบคุมปริมาณแสงสว่างให้เหมาะสมกับลักษณะการใช้งานที่แตกต่างกันของผู้ใช้งาน โดยพัฒนาการใช้หุ่นจำลองร่วมกับซอฟต์แวร์ช่วยวิเคราะห์ระบบแสงสว่างเพื่อการออกแบบอนุรักษ์พลังงาน โดยตัวแปรเชิงปริมาณที่นำมาพิจารณามี 3 ตัวแปร ดังนี้

1. ตำแหน่งและจำนวนหลอดไฟ (ผังการจัดวาง)
2. ค่าความส่องสว่าง (ลักซ์)
3. ค่าการใช้พลังงาน (บาท)

ส่วนข้อปลีกย่อยที่ไม่ได้นำมาพิจารณา ได้แก่

1. การควบคุมค่าความส่องสว่างนี้ จำเป็นต้องสร้างโมเดลเพื่อจำลองการทำงานของฮาร์ดแวร์ ดังนั้นขนาดของห้องจำเป็นต้องคงที่ไม่สามารถปรับเปลี่ยนได้

2. การคำนวณค่าความส่องสว่าง นำมาพิจารณาเพียงระนาบพื้น โดยไม่คิดคำนวณค่าความส่องสว่างจากฟลั๊กซ์ส่องสว่างที่สะท้อนไปมา ถึงแม้ว่าวิธีรับค่าแสงจากตัวเซนเซอร์ที่ใช้อยู่ในซอฟต์แวร์ ต้องใช้ค่าการสะท้อนของพื้นเพดานและผนังก็ตาม เนื่องจากวิธีการคำนวณค่อนข้างซับซ้อนและละเอียด

3. การจัดวางตำแหน่งดวงโคมและการควบคุมการส่องสว่าง โดยพิจารณาปริมาณค่าความส่องสว่างที่เหมาะสมในพื้นที่หนึ่ง ๆ เท่านั้น และไม่พิจารณาเรื่องของความสวยงามเข้ามาเกี่ยวข้อง

4. การจัดวางตำแหน่งหลอดไฟ เลือกใช้หลอดไฟประเภทเดียวในพื้นที่นั้น

วิธีการควบคุมเป็นรูปแบบของซอฟต์แวร์คอมพิวเตอร์ และเพื่อสะดวกในการใช้งานคอมพิวเตอร์สามารถคำนวณสมการที่มีความซับซ้อนได้อย่างรวดเร็ว ซอฟต์แวร์สามารถใช้กับคอมพิวเตอร์ส่วนบุคคลที่มีอยู่ในทุกสำนักงานหรือคอมพิวเตอร์โน้ทบุ๊ค (notebook) ซึ่งสามารถพกพาได้ผู้ใช้งานไม่จำเป็นต้องมีความรู้เรื่องการจัดการระบบแสงสว่าง ทำให้สามารถเข้าใจผลลัพธ์การควบคุมได้ เพียงทราบข้อมูลเบื้องต้น แล้วเลือกหรือปรับค่าความสว่างในซอฟต์แวร์คอมพิวเตอร์จากนั้นผลลัพธ์ก็จะแสดงตามที่ต้องการ เพื่อนำไปวิเคราะห์เปรียบเทียบ ช่วยในการออกแบบและวิเคราะห์ปรับปรุงระบบแสงสว่างเพื่อประสิทธิภาพการใช้พลังงานได้อย่างรวดเร็ว

3.2 กระบวนการในการพัฒนาซอฟต์แวร์และฮาร์ดแวร์

สามารถอธิบายรายละเอียดได้ดังนี้

1. กำหนดขอบเขตการประมวลผลของซอฟต์แวร์ แล้วนำมาศึกษาวิเคราะห์ปรับปรุงระบบควบคุมค่าความส่องสว่างโดยกำหนดแนวทางความคิดที่ใช้ในการออกแบบซอฟต์แวร์

   1) การออกแบบลักษณะการแสดงผลของซอฟต์แวร์
   2) เทคนิควิธีการที่ใช้ในการควบคุมค่าความสว่างภายในห้อง
   3) เทคนิควิธีการที่เชื่อมต่อซอฟต์แวร์และฮาร์ดแวร์ไมโครคอนโทรลเลอร์
   4) การแสดงผลของการควบคุม

2. ออกแบบเรียบเรียงการเชื่อมโยงของซอฟต์แวร์ และส่วนประกอบของฮาร์ดแวร์เข้าด้วยกัน เพื่อให้สะดวกและเหมาะสมกับการใช้ควบคุมฮาร์ดแวร์ จากนั้นลำดับการใช้งานให้ผู้ใช้ซอฟต์แวร์ใช้งานได้ง่าย

3. ทำแบบจำลองโมเดลห้อง เพื่อทดลองติดตั้งหลอดไฟ และควบคุมค่าความส่องสว่างผ่านฮาร์ดแวร์ด้วยซอฟต์แวร์ที่ได้พัฒนาขึ้น

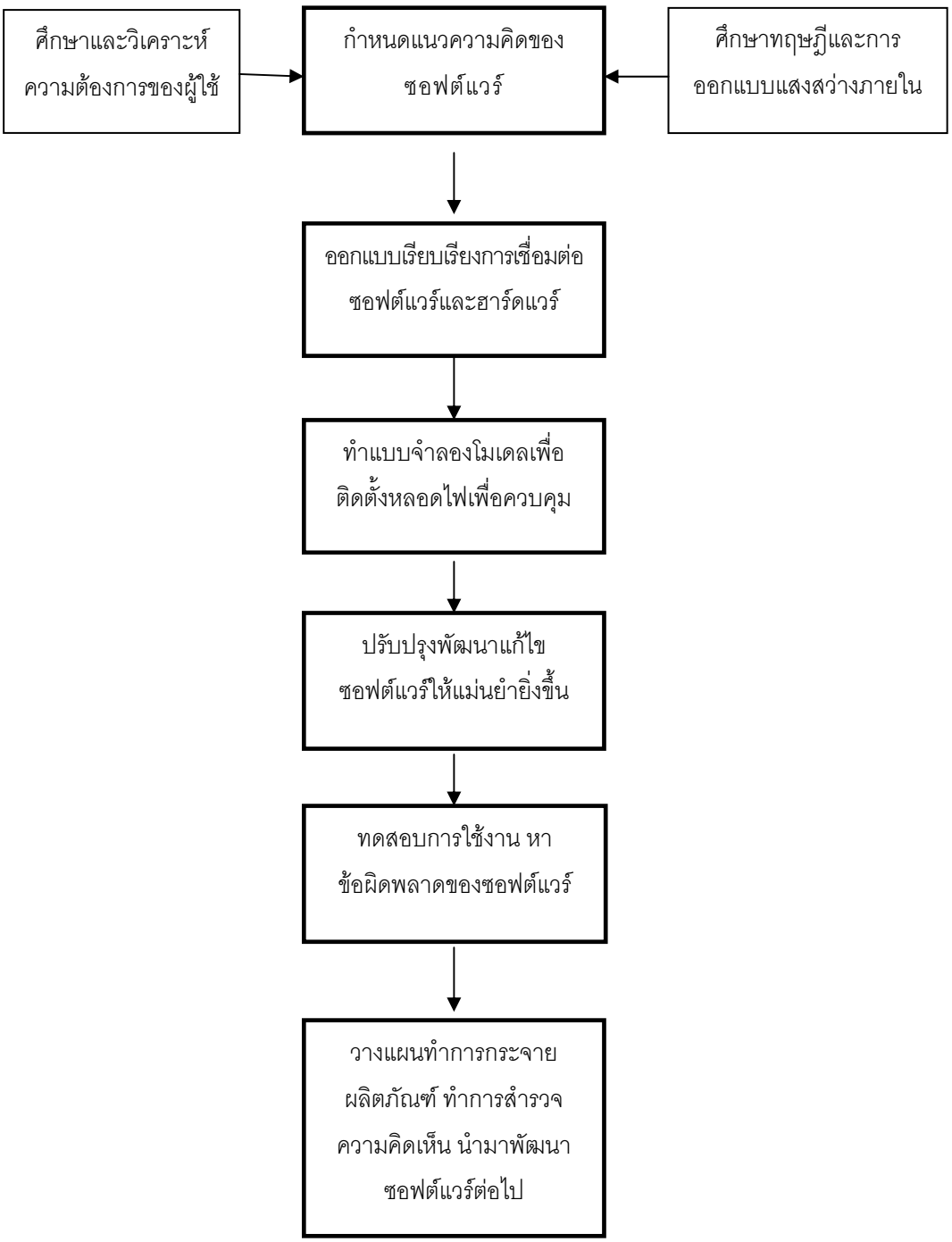
4. ปรับปรุงพัฒนาเนื้อหาแก้ไขซอฟต์แวร์ และทำการปรับปรุงซอฟต์แวร์ให้ควบคุมฮาร์ดแวร์เพื่อค่าความส่องสว่างที่มีความถูกต้องมากยิ่งขึ้น

5. ทดสอบการใช้งาน หาข้อผิดพลาดของซอฟต์แวร์ และทดสอบการเชื่อมต่อระหว่างซอฟต์แวร์และฮาร์ดแวร์ เพื่อลดความผิดพลาดที่เกิดจากการใช้ซอฟต์แวร์ให้มากที่สุด โดยอาจทดลองเปรียบเทียบกับผลที่ได้จากการใช้ซอฟต์แวร์เกี่ยวกับแสงสว่างคำนวณออกมา ว่าซอฟต์แวร์สามารถปรับระดับแสงสว่างได้เหมาะสม และสัมพันธ์กับสภาพของแสงธรรมชาติที่มีอยู่หรือติดตั้งหลอดไฟที่หน้างานจริงโดยใช้ฮาร์ดแวร์ควบคุมปริมาณความสว่าง วัดค่าแสงสว่างที่เกิดขึ้น แล้วปรับความสว่างให้เหมาะสมตามค่าที่วัดได้จริง

6. วางแผนกระจายผลิตภัณฑ์สู่ท้องตลาด หลังจากที่พัฒนาซอฟต์แวร์ให้มีความแม่นยำ และเหมาะกับการใช้งานสำหรับผู้ใช้งานที่หลากหลาย แล้วทำการสำรวจความคิดเห็นของผู้ใช้งาน เพื่อนำมาพัฒนาซอฟต์แวร์ต่อไป

ภาพที่ 3.1

กระบวนการในการพัฒนาซอฟต์แวร์เพื่อควบคุมฮาร์ดแวร์



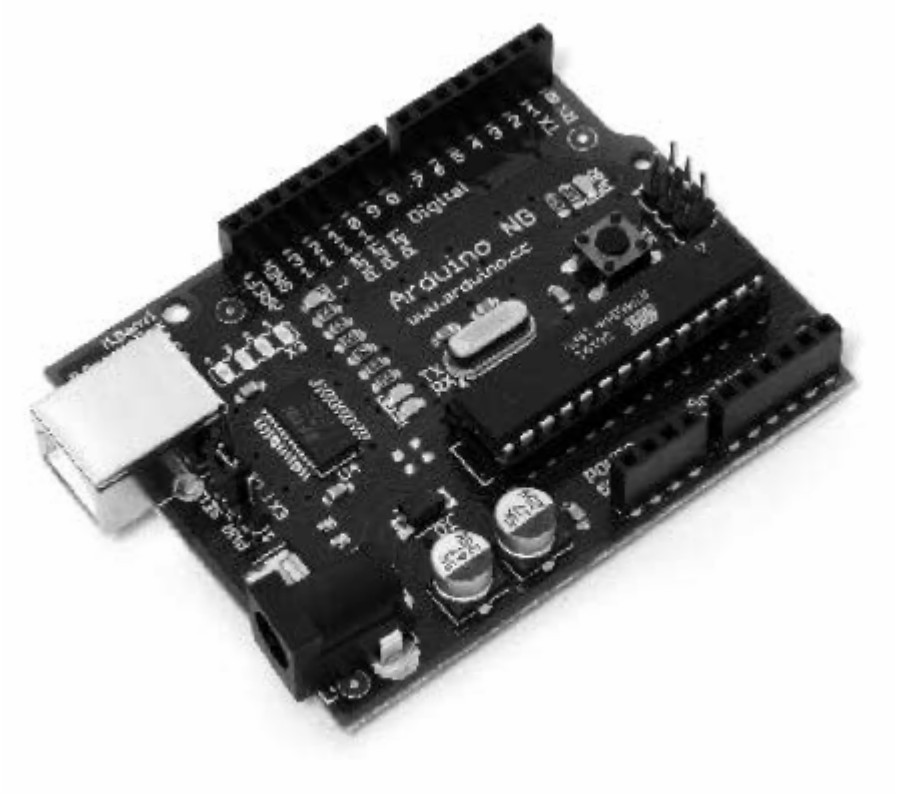
3.3 การติดตั้งอุปกรณ์ที่ใช้ควบคุมค่าความสว่าง

จากงานวิจัยนี้ซอฟต์แวร์ที่ถูกพัฒนาขึ้นจำเป็นต้องติดตั้งอุปกรณ์หลายอย่าง จึงจะสามารถควบคุมค่าความสว่างได้ รวมทั้งต้องสร้างโมเดลจำลองการติดตั้งหลอดไฟ เพื่อดูผลลัพธ์ของการควบคุมและแก้ไขส่วนที่ผิดพลาด ซึ่งอุปกรณ์ที่ใช้ในการติดตั้งมีดังนี้

1. ฮาร์ดแวร์ไมโครคอนโทรลเลอร์ เป็นอุปกรณ์อย่างแรกที่ ต้องคำนึงถึงในงานวิจัยนี้ ซึ่งในการพัฒนาซอฟต์แวร์ ผู้วิจัยได้เลือกใช้ไมโครคอนโทรลเลอร์ของยี่ห้ออาร์ดูอีโน่ โดยทำการเชื่อมต่อจากคอมพิวเตอร์ด้วยสายยูเอสบี (USB) เพราะการส่งค่าคำสั่งการใช้งานจากซอฟต์แวร์ในคอมพิวเตอร์นั้นเป็นการส่งมาเป็นคำสั่ง ซีเรียลพอร์ท (Serial Port)

ภาพที่ 3.2

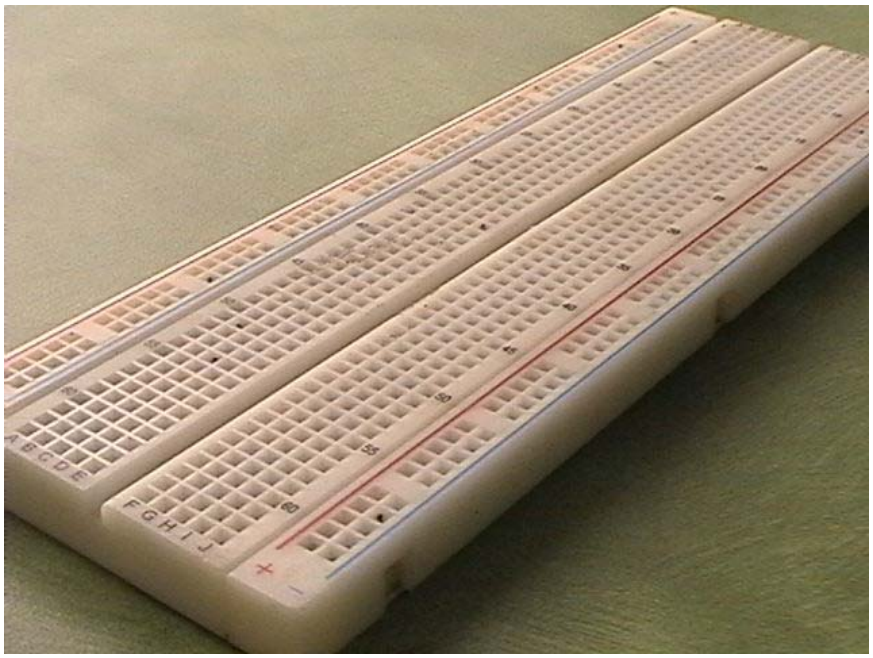
ฮาร์ดแวร์ไมโครคอนโทรลเลอร์อาร์ดูอีโน่ที่ใช้ในงานวิจัย



หมายเหตุ: ถ่ายภาพโดยผู้วิจัย 16 ธันวาคม พ.ศ. 2551.

2. เบรดบอร์ด เป็นอุปกรณ์ที่มีลักษณะเป็นรู ทำหน้าที่ในการเชื่อมต่อระบบการควบคุมทั้งหมดจากไมโครคอนโทรลเลอร์ผ่านสายไฟและตัวต้านทานขนาดต่าง ๆ รวมไปถึงการเพิ่มจำนวนหลอดไฟที่จะติดตั้งได้ โดยการใช้สายไฟปักเข้าไปในรูของตัวเบรดบอร์ดตามจำนวนที่ต้องการ แต่ทั้งนี้ต้องไม่เกินกว่าจำนวนรูของเบรดบอร์ด ซึ่งมีอยู่หลายรู

ภาพที่ 3.3
เบรดบอร์ดที่ใช้ในการเชื่อมต่อระบบควบคุมทั้งหมด



หมายเหตุ: ถ่ายภาพโดยผู้วิจัย 16 ธันวาคม พ.ศ. 2551.

3. โมเดลถือว่าเป็นอุปกรณ์ชิ้นหนึ่งเหมือนกัน เพราะระบบควบคุมจากซอฟต์แวร์ผ่านไมโครคอนโทรลเลอร์นั้น ต้องมีการจำลองติดตั้งหลอดไฟ และตัวเซนเซอร์เพื่อแสดงให้เห็นผลลัพธ์ของการควบคุม

ภาพที่ 3.4
โมเดลที่ใช้จำลองติดตั้งหลอดไฟในงานวิจัยนี้

หมายเหตุ: ถ่ายภาพโดยผู้วิจัย 18 มีนาคม พ.ศ. 2552.

4. เซนเซอร์วัดปริมาณแสง ซึ่งในงานวิจัยนี้ใช้เซนเซอร์แอลดีอาร์ (LDR) 6 ตัว โดยการติดตั้งนั้น จะใช้สายไฟต่อมาจากเบรดบอร์ดที่ถูกเชื่อมต่อกับไมโครคอนโทรลเลอร์แล้ว จากนั้นนำไปใส่ไว้ในโมเดลห้องที่ทำจำลองขึ้นมาเพื่อวัดค่าแสงสว่างภายในโมเดล

ภาพที่ 3.5

เซนเซอร์แอลดีอาร์ที่ใช้ในงานวิจัย

หมายเหตุ: ถ่ายภาพโดยผู้วิจัย 16 ธันวาคม พ.ศ. 2551.

5. สวิทช์เปิด – ปิด เนื่องจากในการพัฒนาซอฟต์แวร์มีข้อจำกัด ในการแยกส่วนของการทำงานแบบอัตโนมัติและแบบควบคุมเอง จึงแก้ไขโดยใช้สวิทช์เปิด – ปิด มาเป็นตัวกำหนดและปรับเปลี่ยนรูปแบบการทำงาน โดยทำการติดตั้งไว้ที่เบรดบอร์ด

6. หลอดไฟแอลอีดี (LED) ชนิดที่ให้ความสว่างมากกว่าหลอดไฟแอลอีดีปกติ ซึ่งงานวิจัยนี้จำเป็นต้องใช้นำมาติดตั้งเพื่อจำลองความสว่างภายในโมเดลที่ทำจำลองขึ้นมา โดยการใช้สายไฟ และตัวต้านทานเชื่อมต่อจากไมโครคอนโทรลเลอร์ เพื่อทำการควบคุมจากซอฟต์แวร์

ภาพที่ 3.6

หลอดไฟแอลอีดีแบบสว่างมาก

หมายเหตุ: ถ่ายภาพโดยผู้วิจัย 16 ธันวาคม พ.ศ. 2551

การพัฒนาซอฟต์แวร์เพื่อควบคุมไมโครคอนโทรลเลอร์ในงานวิจัยนี้ มีความซับซ้อนในการติดตั้งอุปกรณ์ดังที่ได้กล่าวมาแล้ว เนื่องจากการพัฒนาซอฟต์แวร์จำเป็นต้องมีการทดลองให้

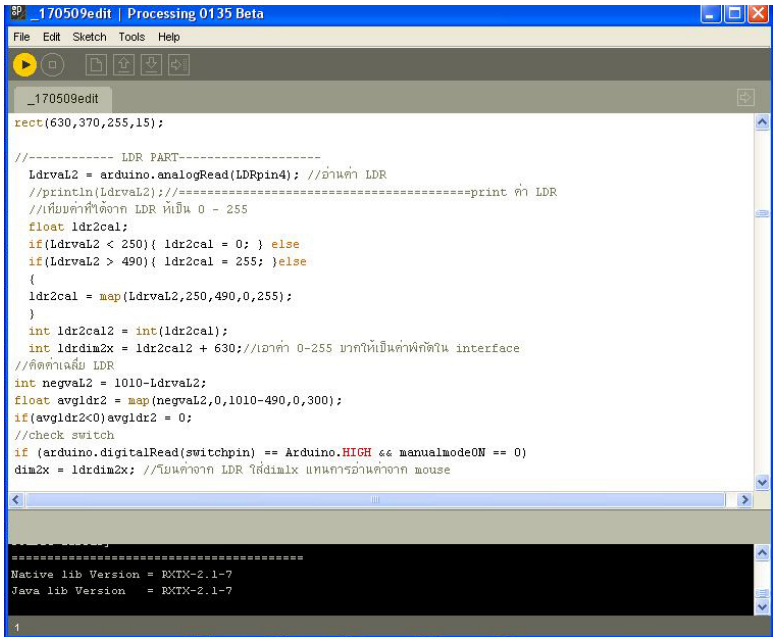
เห็นถึงผลลัพธ์ของการควบคุมจากซอฟต์แวร์ที่พัฒนาขึ้น อุปกรณ์ที่ใช้จึงมีหลายอย่าง ซึ่งแต่ละอย่างอาจไม่เป็นที่รู้จักสำหรับสถาปนิกหรือบุคคลทั่วไป แต่การเลือกอุปกรณ์ต่าง ๆ ที่ใช้ในงานวิจัยนี้ได้มีการศึกษามาเป็นอย่างดีแล้วจากเอกสารอ้างอิง

ผลที่ได้จากการศึกษา สามารถศึกษาเพิ่มเติมในส่วนของปริมาณการส่องสว่างของหลอดไฟและแสงธรรมชาติจากช่องเปิด รวมถึงความต้องการความส่องสว่างในห้องประเภทต่าง ๆ เพื่อให้ทราบเทคนิคที่เหมาะสม และประยุกต์ใช้กับการใช้งานต่อไป

3.4 การแสดงผลการควบคุมค่าความสว่าง

งานวิจัยนี้เลือกใช้ซอฟต์แวร์ในการพัฒนาเป็นส่วนสำคัญที่จะชี้แนวทางและลักษณะเด่น (feature) ของซอฟต์แวร์ โดยผู้เขียนเลือกใช้ซอฟต์แวร์โปรเซสซิ่ง (Processing) ในส่วนของการทำระบบติดต่อกับผู้ใช้ (User Interface) และใช้ซอฟต์แวร์อาร์ดูอีโน่ (Arduino) ในส่วนของการทำซอฟต์แวร์ควบคุมไมโครคอนโทรลเลอร์ อีกทั้งซอฟต์แวร์โปรเซสซิ่ง มีความสามารถพิเศษในการเชื่อมต่อกับซอฟต์แวร์อาร์ดูอีโน่ ที่ใช้ควบคุมไมโครคอนโทรลเลอร์เหนือกว่าการเขียนด้วยซอฟต์แวร์อื่น ๆ ดังนี้

ภาพที่ 3.7
หน้าจอการทำงานของซอฟต์แวร์โปรเซสซิ่ง



3.4.1 ความสามารถในการสร้างงานกราฟฟิก

การเขียนซอฟต์แวร์โดยใช้ภาษาในการควบคุม โดยมากแล้วจะมีความสามารถในการใช้คำสั่งเกี่ยวกับการวาดภาพไม่ค่อยดีนัก แต่การใช้ซอฟต์แวร์โปรเซสซิ่งมีความสามารถในการวาดภาพแบบ 2 มิติได้ดีซึ่งเป็นที่นิยมกันในหมู่นักออกแบบที่สนใจทางด้านคอมพิวเตอร์โปรแกรม อีกทั้งซอฟต์แวร์โปรเซสซิ่งมีความสามารถในส่วนที่เป็นภาษาคอมพิวเตอร์สำหรับแปลงค่าสมการ และยังเหมาะกับผู้ที่ไม่ได้ศึกษาเกี่ยวโปรแกรมเมอร์มาโดยตรง ยิ่งไปกว่านั้นกราฟฟิกที่ถูกทำขึ้นยังสามารถนำมาดัดแปลงให้เป็นส่วนหนึ่งของการทำงานร่วมกับฮาร์ดแวร์ได้ ซึ่งสามารถนำมาแปลงค่าของซอฟต์แวร์ที่จะพัฒนาขึ้นได้ต่อไปอีกในอนาคต แม้ว่าจะเขียนภาษาคอมพิวเตอร์ที่ใช้งานร่วมกับฮาร์ดแวร์ออกมาได้อย่างไม่เต็มรูปแบบนัก แต่ตัวซอฟต์แวร์สามารถเผยแพร่ไปยังสื่อต่าง ๆ ได้ง่าย ผู้วิจัยจึงเลือกทำงานกับซอฟต์แวร์นี้

3.4.2 ความสามารถในการสร้างระบบเชื่อมโยงกับฮาร์ดแวร์ไมโครคอนโทรลเลอร์

ในการใช้ซอฟต์แวร์โปรเซสซิ่งให้สามารถควบคุมฮาร์ดแวร์ไมโครคอนโทรลเลอร์ได้นั้น จำเป็นต้องใช้ซอฟต์แวร์อาร์ดูอีโน่ เป็นตัวสั่งการว่าจะให้ไมโครคอนโทรลเลอร์ทำอะไรได้บ้างตามเงื่อนไขที่กำหนด ซึ่งในการสร้างเงื่อนไขนี้ได้มีการเชื่อมโยงระบบของไมโครคอนโทรลเลอร์ไปสู่ระบบการทำงานของซอฟต์แวร์โปรเซสซิ่งพร้อม ๆ กัน โดยสามารถควบคุมอุปกรณ์ไฟฟ้าได้ทั้งผ่านไมโครคอนโทรลเลอร์ และ บอร์ดออนออฟ (I/O Boards) โดยสามารถแบ่งอุปกรณ์ตัวกลางได้เป็น 3 ประเภทดังนั้น

1. แบร์ไมโครคอนโทรลเลอร์ (Bare Microcontroller) หมายถึง ไมโครคอนโทรลเลอร์ที่ไม่ได้มีแผงวงจรสำเร็จรูป เป็นไมโครคอนโทรลเลอร์เพียงอย่างเดียว ซึ่งได้แก่ ไมโครชิป (Microchip PIC) และ แอ็ทเมล เอวีอาร์ (Atmel AVR) การทำงานกับไมโครคอนโทรลเลอร์ประเภทนี้จะมีความยืดหยุ่นสูง เพราะไม่ได้ยึดติดกับอุปกรณ์บนบอร์ด แต่กระนั้นยังต้องอาศัยความชำนาญสูงด้วยเช่นกัน ไมโครคอนโทรลเลอร์ทั้งสองตระกูลนี้มีหลากหลายรุ่นด้วยกัน ซึ่งมีราคาแปรผันไปตามประสิทธิภาพ แต่โดยปรกติแล้ว เฉพาะไมโครคอนโทรลเลอร์ จะมีราคาถูกกว่าบอร์ดออนออฟ (I/O Boards) มาก

ในด้านของภาษาที่ใช้ในโปรแกรม โดยทั่วไปแล้วจะใช้ภาษาซี (C) แต่ในบางครั้งสามารถโปรแกรมคำสั่งโดยใช้ภาษาอื่น ๆ เช่นภาษาเบสิค (BASIC) ได้เช่นกัน

2. โปรแกรมเอเบิลบอร์ดออนออฟ (Programmable I/O Boards) หมายถึงบอร์ดที่มีไมโครคอนโทรลเลอร์ติดตั้งอยู่บนแผงวงจร พร้อมอุปกรณ์ต่าง ๆ ตัวอย่างบอร์ดประเภทนี้ได้แก่ ไวริ่ง (Wiring), อาร์ดูอีโน่ (Arduino), เบสิคแสตมป์ 2 (Basic Stamp2), บีเอ็กซ์ 24 (BX-24), พิค (PIC)

สำหรับบอร์ดไวริ่ง และ อาร์ดูอีโน่ บอร์ดทั้งสองนี้ได้ออกแบบมาเพื่อการใช้งานของดีไซน์เนอร์ และศิลปินที่สนใจในงานอิเล็กทรอนิกส์ จึงใช้งานได้ง่ายกว่าบอร์ดประเภทอื่น ภาษาที่ใช้ทั้งสองบอร์ดนี้ใช้ภาษาไวริ่ง ในการควบคุมไมโครคอนโทรลเลอร์ และใช้โปรแกรมโปรเซสซิ่งในการเขียนโปรแกรม จึงทำให้ทั้งสองบอร์ดนี้สามารถใช้คำสั่งร่วมกับโปรแกรมโปรเซสซิ่ง ได้ในบางโอกาส เมื่อเปรียบเทียบภาษาไวริ่ง กับภาษาโปรเซสซิ่งแล้ว ทั้งสองภาษานี้มีระดับความยากของภาษาอยู่ในระดับเดียวกัน แต่ภาษาไวริ่งจะมีคำสั่งที่เฉพาะเจาะจงในการควบคุมไมโครคอนโทรลเลอร์ และจะปฏิเสธข้อมูลในด้านกราฟฟิคซึ่งแตกต่างจากภาษาโปรเซสซิ่งที่มีความโดดเด่นในด้านกราฟฟิค อย่างไรก็ตาม ทั้งสองภาษายังมีจุดร่วมในการแปลภาษาก่อนใช้งาน นั่นคือทั้งไวริ่งและโปรเซสซิ่งต้องแปลออกมาเป็นภาษาซีบวกบวก (C/C++) ก่อน แล้วจึงใช้ตัวแปลภาษาซีบวกบวก ในการแปลให้คอมพิวเตอร์ปฏิบัติงาน

3. เททเทอเร็ดบอร์ดออนออฟ (Tethered I/O Board) เป็นบอร์ดที่นิยมใช้ในการรับข้อมูลจากอุปกรณ์ต่าง ๆ เช่น อุปกรณ์รับแสง และเซ็นเซอร์ต่าง ๆ บอร์ดได้ถูกออกแบบมาให้ใช้งานได้ง่ายและอาจไม่ต้องอาศัยความรู้ทางด้านอิเล็กทรอนิกส์ เนื่องจาก อุปกรณ์รับค่าต่าง ๆ เช่น มอเตอร์ อุปกรณ์รับแสง ได้ถูกผนวกติดมากับบอร์ดพร้อมอยู่แล้ว และสามารถส่งค่าไปยังซอฟต์แวร์ต่าง ๆ เช่น แมกซ์เอ็มเอสพี (Max/MSP), แฟรช (Flash), ไดเรคเตอร์ (Director) และซอฟต์แวร์ภาษาอื่น ๆ แต่ในทางกลับกัน บอร์ดประเภทนี้จะมีราคาสูงมากกว่าประเภทอื่น ๆ ตัวอย่างบอร์ดประเภทนี้ได้แก่ เทลีโอ (Teleo) ไอคูปเอ็กซ์ (I-Cube X) ฟิดเจ็ท (Phidgets) อีซิโอ (EZIO) เป็นต้น

3.4.3 ความสามารถในการสร้างระบบติดต่อกับผู้ใช้งาน

ในซอฟต์แวร์โปรเซสซิ่งมีส่วนที่เป็นฟังก์ชั่นตัวอย่างซึ่งเป็นส่วนสำเร็จรูปของซอฟต์แวร์ที่ตัวซอฟต์แวร์ได้ทำตัวอย่างของการเขียนโปรแกรมจากซอฟต์แวร์ไว้ โดยสามารถทำความเข้าใจได้ไม่ยากนัก ซึ่งผู้ใช้งานสามารถใช้หลักการเขียนภาษาคอมพิวเตอร์จากตัวอย่างเหล่านี้ในการสร้างระบบติดต่อกับผู้ใช้งานอย่างสมบูรณ์แบบได้ง่าย โดยไม่ต้องผ่านซอฟต์แวร์อื่น ๆ อีกซึ่งภาษาคอมพิวเตอร์ส่วนมาก ต้องการซอฟต์แวร์มาแปลงข้อมูลใส่ในระบบติดต่อ เพื่อสร้างรูปแบบ

การใช้งานอีกครั้ง ในการแสดงผลการควบคุม สามารถแบ่งกลุ่มของซอฟต์แวร์ออกได้เป็น 2 กลุ่ม ดังนี้

1. การแสดงผลแบบกราฟฟิค มักใช้ในการวิเคราะห์ต่าง ๆ เช่น ซอฟต์แวร์อาจเป็น 2 มิติ ใช้ความสามารถของเครื่องไม่สูงมากนัก สามารถแสดงผลในลักษณะการวิเคราะห์ ได้แก่ การวิเคราะห์ค่าความสว่างที่ตัวเซนเซอร์ (LDR) วัดได้แล้วปรับค่าความสว่างได้สัมพันธ์เหมาะสมบนพื้นที่โมเดลห้องที่จำลองขึ้นมา การวิเคราะห์ค่าใช้จ่าย และค่าพลังงานไฟฟ้าที่ใช้ไป แล้วนำค่าดังกล่าวมาเปรียบเทียบประสิทธิภาพกับการใช้แสงสว่างในสถานการณ์จริง

2. การแสดงผลแบบ real-time interactive ได้เพราะซอฟต์แวร์โปรเซสซิ่งทำงานสัมพันธ์กับไมโครคอนโทรลเลอร์อาร์ดูอีโน่ จึงทำให้สามารถเห็นถึงผลที่เกิดขึ้นได้ทันทีตามสภาพแวดล้อมในขณะนั้น ซึ่งวิธีการใช้งานร่วมกับฮาร์ดแวร์มีดังนี้

1) คัดลอกโค้ดที่สามารถช่วยให้ซอฟต์แวร์โปรเซสซิ่งส่งค่าผ่านยูเอสบีไปยังไมโครคอนโทรลเลอร์ได้ โดยคัดลอกไปไว้ในส่วนของการเขียนซอฟต์แวร์ ซึ่งโค้ดดังกล่าวนั้นสามารถหาได้ในอินเตอร์เนท มีชื่อว่า สแตนดาร์ดเฟิมมาต้า (Standard Firmata)

2) ทำการเขียนภาษาคอมพิวเตอร์โปรเซสซิ่งตามหลักการที่วางไว้นั่นคือ อัลกอริทึม (Algorithm) ซึ่งจากการคัดลอกโค้ดสแตนดาร์ดเฟิมมาต้า ทำให้การเขียนง่ายขึ้น สามารถเขียนไปได้เลยว่า ต้องการให้ไมโครคอนโทรลเลอร์อาร์ดูอีโน่ทำอะไรบ้าง ตามหลักของการเขียนภาษาโปรเซสซิ่ง

เนื้อหาที่กล่าวไปข้างต้นเป็นแนวทางที่นำมาพัฒนาซอฟต์แวร์เพื่อให้บรรลุตามวัตถุประสงค์ ซึ่งมีแนวความคิดเป็นตัวกำหนดแนวทางว่าซอฟต์แวร์ควรมีการแสดงผลเป็นอย่างไร จึงจะตอบสนองความต้องการของผู้ใช้งานมากที่สุด

3.5 โครงสร้างของซอฟต์แวร์

จากกระบวนการพัฒนาซอฟต์แวร์สามารถกำหนดแนวคิดของซอฟต์แวร์ต่อไป โดยทำการกำหนดโครงสร้างซอฟต์แวร์ โดยสรุปได้ดังนี้

1. ผู้ใช้งานทำการเปิดใช้หลอดไฟที่ติดตั้งในโมเดลจำลองจากตัวซอฟต์แวร์ ซึ่งค่าแสงสว่างที่วัดได้ภายในโมเดลจำลองจะมีค่าใกล้เคียง 300 ลักซ์ แม้ว่าสภาพแวดล้อมจะเป็นอย่างไรก็ตาม เนื่องจากเซนเซอร์ที่ติดตั้งอยู่ภายในโมเดลจะวัดค่าแสงธรรมชาติที่มีอยู่ในขณะนั้น

แล้วจะส่งค่าไปยังไมโครคอนโทรลเลอร์เพื่อทำการดิมไฟให้สัมพันธ์กับสภาพแวดล้อมที่ควรจะเป็น ซึ่ง 300 ลักซ์เป็นค่าที่เหมาะสมกับขนาดของห้องที่จำลองขึ้นมา

2. ในกรณีที่ผู้ใช้งานต้องการใช้งานในกิจกรรมอื่น ๆ ซอฟต์แวร์จะมีชุดคำสั่งให้เลือกใช้ดังนี้

1) คำสั่งหลัก (MODE) แบ่งเป็น อัตโนมัติ (Auto) และ ปรับด้วยตัวเอง (Manual) คือการเลือกว่าจะให้ซอฟต์แวร์ประมวลผลจากตัวเซนเซอร์ หรือผู้ใช้งานจะควบคุมซอฟต์แวร์ด้วยตัวเอง

2) คำสั่งเลือกใช้ค่าความสว่างลักษณะต่าง ๆ กัน (LIGHTING SCHEME) คือการเลือกระดับปริมาณแสงที่ผู้ใช้งานต้องการ ตามที่ซอฟต์แวร์ตั้งค่าไว้ มี 3 ค่าได้แก่ ความสว่างน้อย (Gloomy), ความสว่างปานกลาง (Warm), ความสว่างสูงสุดที่เหมาะสมกับห้องทดลองในงานวิจัย (Bright)

3) คำสั่งเลือกลักษณะของการเปิดช่องแสง (Type: Windows) เป็นคำสั่งที่ใช้ในการเลือกว่าเมื่อหน้าต่างช่องไหนเปิด ผู้ใช้งานก็ควรจะเลือกใช้ชุดคำสั่งให้ถูกต้องตรงกับหน้าต่างที่เปิดนั้น เพื่อให้แสงสว่างเกิดความเหมาะสมกับสภาพการใช้งาน ซึ่งลักษณะการเปิดมี 3 แบบ คือ แบบที่ 1 เปิดทิศตะวันตก, แบบที่ 2 เปิดทางทิศเหนือ และแบบที่ 3 เปิดหมดทั้ง 2 ทิศ

4) คำสั่งปรับระดับค่าความสว่างอย่างละเอียด (Dimmer) คือคำสั่งที่ใช้ปรับค่าความสว่างของหลอดไฟแต่ละดวงด้วยตัวของผู้ใช้งาน โดยมีหน่วยเป็นเปอร์เซนของค่าความสว่างที่หลอดไฟแสดงออกมา

3. เมื่อผู้ใช้งานเลือกใช้ชุดคำสั่งจนได้ค่าความสว่างตามต้องการแล้ว ถ้าผู้ใช้งานพอใจกับค่าความสว่างนั้น ซอฟต์แวร์จะมีชุดคำสั่งให้เลือกเพื่อเซฟเก็บค่านั้นไว้ใช้ต่อได้ แต่ถ้ายังไม่พอใจกับค่าความสว่าง ก็สามารถกลับไปที่ชุดคำสั่งดิมเมอร์ เพื่อปรับค่าความสว่างให้ได้ตามความต้องการแล้วทำการเซฟค่าเพื่อเก็บไว้ใช้งานต่อไป

4. หลังจากเลือกชุดคำสั่ง และปรับแต่งค่าความสว่างได้ตามต้องการแล้ว ซอฟต์แวร์จะแสดงค่าความสว่างของหลอดไฟทั้งที่ใช้งานในห้องขณะนั้น เพื่อนำไปเปรียบเทียบกับค่าความสว่างภายในทั้งหมดของห้องที่จำลองขึ้นมา

5. ซอฟต์แวร์จะเปรียบเทียบการใช้พลังงานโดยการแสดงผลเป็นกราฟฟิคอย่างง่าย เพื่อให้ผู้ใช้งานสามารถพิจารณาได้ว่าการใช้พลังงานมีประสิทธิภาพหรือไม่และเสียค่าพลังงานไฟฟ้าไปกี่หน่วย (Unit)

6. กำหนดแนวทางการพัฒนาซอฟต์แวร์ในอนาคต เพื่อพัฒนาซอฟต์แวร์ให้มีความสมบูรณ์มากขึ้น

ภาพที่ 3.8

โครงสร้างการทำงานของซอฟต์แวร์