การพัฒนาโปรแกรมสร้างมโนภาพเชิงโต้ตอบด้วยตัวรับรู้ท่าทางสำหรับระบบจัดการพลังงานในอาคารบนพื้นฐาน IEEE1888

นายปรีชา ขาวสะอาด

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2557

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย
บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ที่ส่งผ่านทางบัณฑิตวิทยาลัย
The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository(CUIR)
are the thesis authors' files submitted through the Graduate School.

DEVELOPMENT OF INTERACTIVE VISUALIZATION PROGRAM WITH GESTURE SENSOR FOR IEEE1888-BASED BUILDING ENERGY MANAGEMENT SYSYTEM

MR. PREECHA KHAWSA-ARD

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Electrical Engineering
Department of Electrical Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2014
Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การพัฒนาโปรแกรมสร้างมโนภาพเชิงโต้ตอบ ด้วยตัวรับรู้ท่าทางสำหรับระบบจัดการ พลังงานในอาคารบนพื้นฐาน IEEE1888
โดย	นายปรีชา ขาวสะอาด
สาขาวิชา	วิศวกรรมไฟฟ้า
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร.เชาวน์ดิศ อัศวกุล

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร. บัณฑิต เอื้ออาภรณ์)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.แนบบุญ หุนเจริญ)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ผู้ช่วยศาสตราจารย์ ดร.เชาวน์ดิศ อัศวกุล)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ชัยเชษฐ์ สายวิจิตร)

..... กรรมการภายนอกมหาวิทยาลัย
(ผู้ช่วยศาสตราจารย์ ดร.กิตติพันธุ์ เตชะกิตติโรจน์)

ปรีชา ขาวสะอาด : การพัฒนาโปรแกรมสร้างมโนภาพเชิงโต้ตอบด้วยตัวรับรู้ท่าทางสำหรับระบบจัดการพลังงานในอาคารบนพื้นฐาน IEEE1888 (DEVELOPMENT OF INTERACTIVE VISUALIZATION PROGRAM WITH GESTURE SENSOR FOR IEEE1888-BASED BUILDING ENERGY MANAGEMENT SYSYTEM) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ผศ. ดร. เชาวน์ดิศ อัศวกุล, 98 หน้า.

วิทยานิพนธ์ฉบับนี้นำเสนอการพัฒนาโปรแกรมสร้างมโนภาพเชิงโต้ตอบด้วยตัวรับรู้ท่าทางสำหรับระบบจัดการพลังงานในอาคารบนพื้นฐานของมาตรฐาน IEEE1888 ในโครงการวิจัย และพัฒนาเทคโนโลยีโครงข่ายไฟฟ้าอัจฉริยะเพื่อบริหารจัดการการใช้พลังงานของอาคาร (CU-BEMS) ของภาควิชาวิศวกรรมไฟฟ้า ในวิทยานิพนธ์ฉบับนี้ใช้โปรแกรม processing ในการพัฒนาส่วนแสดงผลหลัก และใช้คลัง(โปรแกรม) SimpleOpenNi ตรวจจับและติดตามโครงกระดูกของผู้ใช้งานเพื่อรับอินพุตจากผู้ใช้งานสำหรับการโต้ตอบกับโปรแกรมที่พัฒนาขึ้น การสื่อสารระหว่างโปรแกรมที่พัฒนาขึ้นกับหน่วยเก็บข้อมูลของโครงการได้นำโพรโทคอล FETCH มาใช้ในการร้องขอข้อมูลการใช้พลังงานเพื่อนำมาแสดงผล และใช้โพรโทรคอล WRITE เพื่อส่งข้อมูลพฤติกรรมการใช้งานของผู้ใช้งานไปเก็บไปในหน่วยเก็บข้อมูล โปรแกรมที่พัฒนาขึ้นถูกติดตั้งร่วมกับหน้าจอขนาด 42 นิ้วพร้อมทั้งตัวรับรู้ท่าทาง kinect เพื่อการใช้งานจริงในโครงการบริเวณ 3 จุด คือ บริเวณหน้าโถงลิฟท์ของชั้น 12 อาคารเจริญวิศวกรรม (ตึก4), บริเวณหน้าโถงลิฟท์ของชั้น 13 อาคารเจริญวิศวกรรม (ตึก4) และบริเวณอาคารปฏิบัติการไฟฟ้าแรงสูง ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ผลจากการทดสอบเป็นระยะเวลา 1 สัปดาห์ ทำให้ทราบพฤติกรรมการใช้งานของผู้ใช้งานโปรแกรมในแต่ละพื้นที่ ซึ่งพบว่าบริเวณหน้าอาคารปฏิบัติการไฟฟ้าแรงสูงเป็นพื้นที่ซึ่งระบบถูกกระตุ้นให้เข้าสู่แคนวาสปริยายจากคนที่เดินผ่านมากที่สุด ซึ่งมี จำนวน 4950 ครั้ง ตามด้วยบริเวณชั้น 13 อาคารเจริญวิศวกรรมซึ่งมีจำนวน 1561 ครั้ง และบริเวณชั้น 12 อาคารเจริญวิศวกรรมซึ่งมีจำนวน 1301 ครั้ง แต่เมื่อวัดความสนใจโดยเริ่มนับจากคนที่เดินผ่านมาแล้วโบกมือ 1 ครั้งขึ้นไปเพื่อโต้ตอบกับหน้าจอแสดงผลข้อมูลพบว่าบริเวณชั้น 12 อาคารเจริญวิศวกรรมได้รับความสนใจสุงสุดโดยคิดเป็นร้อยละ 54.2 บริเวณชั้น 13 อาคารเจริญวิศวกรรมได้รับความสนใจคิดเป็นจำนวนร้อยละ 44.6 และบริเวณอาคารปฏิบัติการไฟฟ้าแรงสูงได้รับความสนใจคิดเป็นจำนวนร้อยละ 37.8 นอกจากนั้นผลจากการทดสอบทำให้ทราบพฤติกรรมการใช้งานของผู้ใช้งานในการโต้ตอบกับโปรแกรมในแคนวาสต่าง ๆ ซึ่งประกอบด้วยแคนวาสสรุปรวมสมรรถนะระบบของภาควิชาวิศวกรรมไฟฟ้า, แคนวาสเกมส์แสดงการใช้พลังงาน, แคนวาสข้อมูลประกาศจากทางภาควิชาวิศวกรรมไฟฟ้า และแคนวาสแจ้งเตือนการใช้พลังงานไฟฟ้า พบว่าผู้ใช้งานมีการโต้ตอบกับโปรแกรมในแคนวาสของเกมส์แสดงการใช้พลังงานมากที่สุดในทุกจุดติดตั้ง ข้อมูลที่ได้จากการทดสอบสามารถนำไปใช้ในปรับปรุงและพัฒนาเสถียรภาพของโปรแกรมให้ทำงานได้อย่างมีประสิทธิภาพมากยิ่งขึ้นในอนาคต นอกจากนั้นจะมีการติดตั้งชุดอุปกรณ์ร่วมกับโปรแกรมที่พัฒนาขึ้นเพิ่มเติมในบริเวณชั้น 1 ของ อาคารภาควิชาวิศวกรรมไฟฟ้าเพื่อเพิ่มพื้นที่ของการนำเสนอข้อมูล

ภาควิชา วิศวกรรมไฟฟ้า **ลายมือชื่อนิสิต**.....
สาขาวิชา วิศวกรรมไฟฟ้า **ลายมือชื่อ อ.ที่ปรึกษาหลัก** .....
ปีการศึกษา .....2557.....

5570278521 : MAJOR ELECTRICAL ENGINEERING
KEYWORDS: DATA VISUALIZATION/ GESTURE SENSOR/ PROCESSING/ IEEE1888/ BUILDING ENERGY MANAGEMENT.

PREECHA KHAWSA-ARD : DEVELOPMENT OF INTERACTIVE VISUALIZATION PROGRAM WITH GESTURE SENSOR FOR IEEE1888-BASED BUILDING ENERGY MANAGEMENT SYSTEM . ADVISOR: ASST. PROF. CHAODIT ASWAKUL, Ph.D., 98 pp.

This thesis presents the development of interactive visualization program with gesture sensor for IEEE1888-based building energy management system of the electrical engineering departmental (CU-BEMS) project. The processing language has been used to develop the main program, the SimpleOpenNI library in the processing environment obtaining the coordinate of user skeletons. This data sent to interpret as events and processed in the processing for control application. The communication protocol in this thesis, FETCH protocol is used to request the data for display on the main program for the protocol to send user behavior data the WRITE protocol has been acquire. The developed program is installed with 42-inch display and kinect at the corridor in front of the elevator on 12 and 13 floor of engineering building 4 as well as in front of the high voltage building, the Department of Electrical Engineering, Faculty of Engineering Chulalongkorn University. After tge test, it is resulted there are the possibility to know the user's behavior of each installation area. The one-week test from three areas shows that the high voltage building is the most woken up spot by number of users passed through total 4950 times. Following, it reveals the area of engineering building 4 (13 fl) by 1561 times and the area of engineering building 4 (12 fl) by 1301 times. In contrast, in terms of user's interests comparing in each area, it shows when measured the attention from passing users and waving their hand greater than or equal to 1 time, the 12 floor area is the most interested spot by the waivers total 54.2 %, the 13 floor area 44.6 % and the high voltage building 37.8 % relatively. The user's behavior on each canvas is another test topic of the study. The developed program includes four main canvas; (1) EE health pad (2) energy game (3) EE information (4) alarm & alert. Among all canvas, the most interested is "energy game". All data of the user's behavior can be used for further development. Finally, the developed system will be installed to the 1st floor of the electrical engineering department building.

Department	: Electrical Engineering	**Student's Signature**	.....
Field of Study	: Electrical Engineering	**Advisor's Signature**	.....
Academic Year	: 2014		

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความช่วยเหลืออย่างดียิ่ง จากอาจารย์ที่ปรึกษาวิทยานิพนธ์ ผศ. ดร.เชาวน์ดิศ อัศวกุล ซึ่งได้ให้ความรู้และคำแนะนำอันมีค่ายิ่งต่อผู้วิจัย อีกทั้งตรวจทานงานวิทยานิพนธ์ฉบับนี้ด้วยดีเสมอมา ผู้วิจัยจึงขอกราบขอบพระคุณมา ณ ที่นี้

งานวิจัยชิ้นนี้ได้รับทุนสนับสนุนจากโครงการขับเคลื่อนการวิจัย กองทุนรัชดาภิเษกสมโภช (Special Task Force for Activating Research (STAR)) ภายใต้กลุ่มวิจัยโครงข่ายไร้สายและอินเทอร์เน็ตอนาคต (Wireless Network and Future Internet Research Group) จุฬาลงกรณ์มหาวิทยาลัย

ขอขอบคุณกลุ่มวิจัย Wireless Network and Future Internet Research Group (WI-FUN) ซึ่งดูแลโดย ผศ. ดร. เชาวน์ดิศ อัศวกุล และ ผศ. ดร. ชัยเชษฐ์ สายวิจิตร ที่จัดกิจกรรมเพื่อส่งเสริมการเรียนรู้และการทำงานของผู้วิจัยให้มีประสิทธิภาพที่ดียิ่งขึ้น รวมถึงให้ความอนุเคราะห์อุปกรณ์เครื่องมือในการทำงานแก่ผู้วิจัย ทำให้งานวิทยานิพนธ์นี้สำเร็จได้อย่างสะดวกราบรื่น

ขอขอบคุณโครงการ CU-BEMS ของภาควิชาวิศวกรรมไฟฟ้า ที่ทำให้ผู้วิจัยด้ความรู้จากการทำงานบนอุปกรณ์จริง รวมทั้งอาจารย์ที่อยู่ในโครงการ ซึ่งได้ให้ความรู้ คำแนะนำอันมีค่า

ขอขอบคุณเพื่อนพี่น้องนักวิจัยทุกคน รวมถึงเจ้าหน้าที่ บุคลากรที่อยู่ในภาควิชาไฟฟ้า สาขาโทรคมนาคม จุฬาลงกรณ์มหาวิทยาลัย ที่ได้ให้ความช่วยเหลือในเรื่องต่าง ๆ และเป็นกำลังใจที่ดียิ่งต่อผู้วิจัย

สุดท้ายนี้ขอขอบคุณครอบครัวของผู้วิจัย ซึ่งได้ให้การสนับสนุนและเป็นกำลังใจให้แก่ผู้วิจัยเสมอมาจนสำเร็จการศึกษา

สารบัญ

หน้า

บทคัดย่อภาษาไทย . . . . . ง
บทคัดย่อภาษาอังกฤษ . . . . . จ
กิตติกรรมประกาศ . . . . . ฉ
สารบัญ . . . . . ช
สารบัญตาราง . . . . . ซ
สารบัญรูป . . . . . ฌ
บทที่
1 บทนำ 1
1.1 ความเป็นมาและความสำคัญของปัญหา 1
1.2 วัตถุประสงค์ของงานวิทยานิพนธ์ 2
1.3 ขอบเขตวิทยานิพนธ์ 2
1.4 ขั้นตอนและวิธีการดำเนินงาน 2
1.5 ประโยชน์ที่คาดว่าจะได้รับ 3
1.6 ประมวลวิทยานิพนธ์ 3
2 หลักการและงานวิจัยในอดีตที่เกี่ยวข้อง 4
2.1 มาตรฐาน IEEE1888 8
2.1.1 สถาปัตยกรรมของ IEEE1888 8
2.1.2 การสื่อสารพื้นฐานในมาตรฐาน IEEE1888 9
2.1.3 การประยุกต์ใช้มาตรฐาน IEEE1888 12
2.2 การสร้างมโนภาพข้อมูลด้วยภาษา processing 13
2.2.1 โปรแกรม processing 13
2.2.2 การประยุกต์ใช้โปรแกรม processing 14
2.3 ตัวรับรู้ท่าทาง (gesture sensor) 15
3 สถาปัตยกรรมระบบและการทดสอบเบื้องต้น 16
3.1 สถาปัตยกรรมของระบบ 16
3.2 การทดสอบเบื้องต้น 17
3.2.1 การทดสอบการทำงานของโปรแกรม processing 17
3.2.2 การทดสอบการทำงานร่วมกันระหว่างตัวรับรู้ท่าทาง kinect และ โปรแกรม processing 20
3.2.3 ทดสอบการเชื่อมต่อระหว่างโปรแกรม processing กับหน่วยเก็บข้อมูลบนพื้นฐาน IEEE1888 โดยใช้โปรแกรม PHP 23
3.2.4 การทดสอบการเชื่อมต่อระหว่างโปรแกรม processing กับหน่วยเก็บข้อมูลบนพื้นฐาน IEEE1888 โดยตรง 26
3.3 การพัฒนา และวิธีการประเมินโปรแกรมต้นแบบ 29
3.3.1 โปรแกรมต้นแบบที่ถูกพัฒนาขึ้น 29
3.3.2 วิธีการประเมินโปรแกรมต้นแบบที่ถูกพัฒนาขึ้น 30

บทที่ หน้า

3.4 ข้อสรุป 31
4 การพัฒนาโปรแกรมสร้างมโนภาพเชิงโต้ตอบด้วยตัวรับรู้ท่าทางตามมาตรฐาน IEEE1888 เพื่อการใช้งานจริงในโครงการ CU-BEMS 32
4.1 การออกแบบโปรแกรมสร้างมโนภาพ 32
4.1.1 รูปแบบการนำเสนอ : ใช้การนำเสนอในเชิงทัศนศิลป์ (visual art) 32
4.1.2 การควบคุมโปรแกรม : ใช้การแสดงท่าทางในการควบคุมการทำงาน 47
4.1.3 การสื่อสารระหว่างโปรแกรมที่พัฒนาขึ้นกับหน่วยเก็บข้อมูลของโครงการ CU-BEMS : ใช้โพรโทคอล FETCH และ WRITE 47
4.2 การทดสอบการใช้งานจริงในโครงการ CU-BEMS 50
4.2.1 การทดสอบเพื่อประเมินพฤติกรรมของผู้เข้าใช้งานระบบ 52
4.2.2 กรณีศึกษา 1 : เปรียบเทียบพฤติกรรมผู้ใช้งานในพื้นที่ติดตั้งแต่ละจุด 57
4.2.3 กรณีศึกษา 2 : เปรียบเทียบพฤติกรรมของผู้ใช้งานในแต่ละแคนวาส (เมื่อมีการตั้งแคนวาสปริยาย) 59
4.2.4 กรณีศึกษา 3 : เปรียบเทียบพฤติกรรมของผู้ใช้งานในแต่ละแคนวาส (เมื่อไม่ตั้งแคนวาสปริยาย) 60
5 บทสรุปและข้อเสนอแนะ 63
5.1 บทสรุป 63
5.2 ข้อเสนอแนะ 64
รายการอ้างอิง 65
ภาคผนวก 67
ก โปรแกรมหลัก (The_Beginning_29092014.pde) 68
ข โปรแกรมเพื่อใช้ในการติดต่อระหว่าง front-end กับ back-end (backend.pde) 74
ค โปรแกรมในส่วนแสดงผลของหน้า screen saver (canvas0.pde, canvas0_init.pde) 78
ง โปรแกรมในส่วนแสดงผลของแคนวาส EE healthpad (canvas1.pde) 79
จ โปรแกรมในส่วนแสดงผลของแคนวาส energy game (canvas2_init.pde,canvas2.pde) 80
ฉ โปรแกรมในส่วนแสดงผลของแคนวาส EE information (canvas3.pde) 86
ช โปรแกรมในส่วนแสดงผลของแคนวาส alarm & alert (canvas4.pde) 88
ซ โปรแกรมในส่วนแสดงผล menu เพื่อใช้เลือกแคนวาส (canvasMenu.pde) 91
ฌ โปรแกรมในส่วนของการโหลดรูปภาพจากโฟลเดอร์เพื่อใช้ในแคนวาสต่าง ๆ (checkImagefile.pde) 94
ฒ โปรแกรมในส่วนแสดงผลนาฬิกา และปฏิทิน (clock.pde) 95
ด โปรแกรมในส่วนของการตรวจจับ และติดตามโครงกระดูกของผู้ใช้งาน (kinect.pde) 96
ประวัติผู้เขียนวิทยานิพนธ์ 98

สารบัญตาราง

หน้า

ตารางที่ 3.1 รายละเอียดของผู้เข้าร่วมทดลองใช้งานโปรแกรมต้นแบบ 30
ตารางที่ 3.2 คุณลักษณะของคอมพิวเตอร์โน๊ตบุ๊คที่ใช้ในการทดสอบโปรแกรมต้นแบบ 30
ตารางที่ 3.3 แบบสอบถามสำหรับประเมินความพึงพอใจโปรแกรมต้นแบบที่ถูกพัฒนาขึ้น . . . 31
ตารางที่ 3.4 ผลคะแนนการแสดงความเห็นของคำถามในแต่ละข้อ 31
ตารางที่ 4.1 บัญชีผู้ใช้ และรหัสผ่านของ dropbox เพื่อใช้บรรจุไฟล์ที่ต้องการประกาศ 40
ตารางที่ 4.2 โควต้าการใช้พลังงาน (kWh) ต่อวันสำหรับแต่ละพื้นที่ของโครงการ CU-BEMS ที่นำมาใช้ในโปรแกรมที่พัฒนาขึ้น 42
ตารางที่ 4.3 รายละเอียดของซอฟต์แวร์ที่ใช้ในการพัฒนาโปรแกรมสร้างมโนภาพ 50
ตารางที่ 4.4 รายละเอียดของฮาร์ดแวร์ที่ใช้ในการพัฒนาโปรแกรมสร้างมโนภาพ 51
ตารางที่ 4.5 จำนวนครั้งที่ระบบถูกกระตุ้นให้เข้าสู่แคนวาสปริยายจากคนที่เดินผ่านใน 1 สัปดาห์ 57
ตารางที่ 4.6 รายละเอียดของหมายเลขแคนวาส 59
ตารางที่ 4.7 จำนวนครั้งที่แคนวาสถูกเรียกใช้งานเมื่อมีการตั้งแคนวาสปริยายไว้ที่หน้า alarm & alert 59
ตารางที่ 4.8 จำนวนครั้งที่แคนวาสถูกเรียกใช้งานเมื่อมีการตั้งแคนวาสปริยายไว้ที่หน้า alarm & alert และกำหนดให้ผู้ใช้งานที่สนใจเรียกใช้แคนวาส alarm & alert คือผู้ใช้งานที่มีการโบกมือตั้งแต่ 2 ครั้งขึ้นไปในการเรียกใช้แต่ละครั้ง 60
ตารางที่ 4.9 จำนวนครั้งที่แคนวาสถูกเรียกใช้งานเมื่อตั้งให้มีการวนแคนวาสปริยาย 60

สารบัญรูป

หน้า

รูปที่ 2.1 โปรแกรมประยุกต์บนเว็บเพื่อเฝ้าสังเกตสภาพอากาศในอาคารบริเวณภาควิชาวิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัย ในงานวิจัย [8], [9] 4
รูปที่ 2.2 ข้อมูลการใช้พลังงานของระบบทดสอบกริดอัจฉริยะในอาคารขนาดเล็กบนเกาะเจจู ประเทศเกาหลีใต้ [10] 5
รูปที่ 2.3 ส่วนต่อประสานกับผู้ใช้ของะบบทดสอบกริดอัจฉริยะในอาคารขนาดเล็กบนเกาะเจจู ประเทศเกาหลีใต้ [10] 5
รูปที่ 2.4 ส่วนต่อประสานกับผู้ใช้ในอาคารชุด Mudeung Park [11] 6
รูปที่ 2.5 แผงจัดการพลังงานสำหรับบ้านอัจฉริยะของโครงการ MeRegioMobil [12] 6
รูปที่ 2.6 โปรแกรมประยุกต์ Greentalk 7
รูปที่ 2.7 ส่วนต่อประสานหลักของโปรแกรม CU-BEMS 7
รูปที่ 2.8 โปรแกรมประยุกต์ CU-BEMS 8
รูปที่ 2.9 สถาปัตยกรรมของมาตรฐาน IEEE1888 [5] 9
รูปที่ 2.10 โพลโทคอล FETCH [5] 10
รูปที่ 2.11 โพลโทคอล WRITE [5] 10
รูปที่ 2.12 โพรโทคอล TRAP [5] 11
รูปที่ 2.13 ตัวรับรู้ และเกตเวย์ที่ใช้โครงการ CU-BEMS 12
รูปที่ 2.14 หน่วยเก็บข้อมูลที่ใช้โครงการ CU-BEMS 12
รูปที่ 2.15 ส่วนต่อประสานหลักของโปรแกรม processing 14
รูปที่ 2.16 ส่วนประกอบของตัวรับรู้ท่าทาง kinect 15
รูปที่ 3.1 สถาปัตยกรรมของระบบเพื่อพัฒนาโปรแกรมสร้างมโนภาพเชิงโต้ตอบ 16
รูปที่ 3.2 โปรแกรมเพื่อทดสอบการดำเนินงาน 17
รูปที่ 3.3 การดำเนินการเมื่อเลือกใช้แบบวิธีจาวา 18
รูปที่ 3.4 การดำเนินการเมื่อเลือกใช้แบบวิธีจาวาสคริปต์ 18
รูปที่ 3.5 ตัวจัดการชุดพัฒนาซอฟต์แวร์แอนดรอยด์ 19
รูปที่ 3.6 การดำเนินการเมื่อเลือกใช้แบบวิธีแอนดรอยด์ 19
รูปที่ 3.7 โปรแกรมทดสอบการตรวจจับผู้ใช้ 20
รูปที่ 3.8 ผลการทดสอบการตรวจจับผู้ใช้ 21
รูปที่ 3.9 โปรแกรมทดสอบการติดตามโครงกระดูก 22
รูปที่ 3.10 โปรแกรมทดสอบการติดตามโครงกระดูก (ต่อ) 22
รูปที่ 3.11 ผลการทดสอบการตรวจจับผู้ใช้ และการติดตามโครงกระดูก 23
รูปที่ 3.12 โปรแกรมทดสอบการร้องขอข้อมูลไปยังหน่วยเก็บข้อมูล 24
รูปที่ 3.13 โปรแกรมทดสอบการร้องขอข้อมูลไปยังหน่วยเก็บข้อมูล(ต่อ) 24
รูปที่ 3.14 โปรแกรมทดสอบการร้องขอข้อมูลไปยังหน่วยเก็บข้อมูล(ต่อ) 25
รูปที่ 3.15 ผลการทดสอบการเชื่อมต่อระหว่างโปรแกรม processing กับหน่วยเก็บข้อมูลบนพื้นฐาน IEEE1888 โดยใช้โปรแกรม PHP 25
รูปที่ 3.16 การร้องขอข้อมูลจาก processing ไปยังหน่วยเก็บข้อมูลโดยโพรโทคอล FETCH . . 26

หน้า

รูปที่ 3.17 การตอบกลับพร้อมทั้งข้อมูลจากหน่วยเก็บข้อมูลเมื่อใช้โพรโทคอล FETCH 27
รูปที่ 3.18 การส่งข้อมูลจาก processing ไปยังหน่วยเก็บข้อมูลเมื่อใช้โพรโทคอล WRITE . . . 28
รูปที่ 3.19 การตอบกลับจากหน่วยเก็บข้อมูลเมื่อใช้โพรโทคอล WRITE 28
รูปที่ 3.20 โปรแกรมต้นแบบ energy game 29
รูปที่ 4.1 หน้าจอแรกของโปรแกรมที่ถูกพัฒนาขึ้น 33
รูปที่ 4.2 หน้าจอสรุปรวมดรรชนีชี้วัดระบบ CU-BEMS บนเว็บไวต์ของโครงการ 33
รูปที่ 4.3 หน้าจอแสดงค่าความต้องการพลังงานสูงสุดที่ลดลงได้จากเกณฑ์สูงสุดที่ตั้งไว้ 34
รูปที่ 4.4 หน้าจอแสดงค่าการใช้พลังงานต่อพื้นที่ 35
รูปที่ 4.5 หน้าจอแสดงค่าโหลดแฟกเตอร์ 35
รูปที่ 4.6 หน้าจอแสดงค่าไฟฟ้าที่ลดได้ 36
รูปที่ 4.7 หน้าจอแสดงค่าพลังงานไฟฟ้าที่ลดได้ 36
รูปที่ 4.8 หน้าจอแสดงค่าพลังงานที่ลดได้เทียบเป็นปริมาณการใช้ถ่านหินสำหรับผลิตไฟฟ้า . . 37
รูปที่ 4.9 หน้าจอแสดงค่าพลังงานที่ลดได้เทียบเป็นปริมาณการปล่อยก๊าซคาร์บอนไดออกไซด์ ในการผลิตไฟฟ้า 37
รูปที่ 4.10 หน้าจอแสดงค่าพลังงานที่จ่ายจากแหล่งพลังงานหมุนเวียนในระบบ 38
รูปที่ 4.11 หน้าจอแสดงค่าพลังงานที่ลดได้เทียบเป็นจำนวนต้นไม้ 38
รูปที่ 4.12 หน้าจอเมื่อเริ่มเกมส์ 39
รูปที่ 4.13 หน้าจอขณะเล่นเกมส์ 39
รูปที่ 4.14 หน้าจอเมื่อสิ้นสุดการเล่นเกมส์ 40
รูปที่ 4.15 หน้าจอแสดงข้อมูลประกาศจากภาควิชา 41
รูปที่ 4.16 หน้าจอแสดงข้อมูลการใช้พลังงานรวมทั้งภาควิชาคิดเป็นร้อยละเมื่อเทียบกับเป้าหมาย 43
รูปที่ 4.17 หน้าจอแสดงข้อมูลการใช้พลังงานของชั้น 2 อาคารภาควิชาวิศวกรรมไฟฟ้า คิดเป็น ร้อยละเมื่อเทียบกับเป้าหมาย 43
รูปที่ 4.18 หน้าจอแสดงข้อมูลการใช้พลังงานของชั้น 3 อาคารภาควิชาวิศวกรรมไฟฟ้า คิดเป็น ร้อยละเมื่อเทียบกับเป้าหมาย 44
รูปที่ 4.19 หน้าจอแสดงข้อมูลการใช้พลังงานของชั้น 4 อาคารภาควิชาวิศวกรรมไฟฟ้า คิดเป็น ร้อยละเมื่อเทียบกับเป้าหมาย 44
รูปที่ 4.20 หน้าจอแสดงข้อมูลการใช้พลังงานของชั้น 5 อาคารภาควิชาวิศวกรรมไฟฟ้า คิดเป็น ร้อยละเมื่อเทียบกับเป้าหมาย 45
รูปที่ 4.21 หน้าจอแสดงข้อมูลการใช้พลังงานของชั้น 12 อาคารเจริญวิศวกรรม คิดเป็นร้อยละ เมื่อเทียบกับเป้าหมาย 45
รูปที่ 4.22 หน้าจอแสดงข้อมูลการใช้พลังงานของชั้น 13 อาคารเจริญวิศวกรรม คิดเป็นร้อยละ เมื่อเทียบกับเป้าหมาย 46
รูปที่ 4.23 หน้าจอแสดงข้อมูลการใช้พลังงานของอาคารไฟฟ้าแรงสูง คิดเป็นร้อยละเมื่อเทียบ กับเป้าหมาย 46
รูปที่ 4.24 โปรแกรมในส่วนของการร้องขอข้อมูลโดยใช้โพรโทคอล FETCH 47
รูปที่ 4.25 โปรแกรมในส่วนของการร้องขอข้อมูลโดยใช้โพรโทคอล FETCH (ต่อ) 48
รูปที่ 4.26 point ID ที่ใช้ในการร้องขอข้อมูล 48
รูปที่ 4.27 โปรแกรมในส่วนของการเขียนข้อมูลโดยใช้โพรโทคอล WRITE 49

หน้า

รูปที่ 4.28 โปรแกรมในส่วนของการเขียนข้อมูลโดยใช้โพรโทคอล WRITE (ต่อ) 49
รูปที่ 4.29 โปรแกรมในส่วนของการเขียนข้อมูลโดยใช้โพรโทคอล WRITE (ต่อ) 50
รูปที่ 4.30 จุดติดตั้งชุดอุปกรณ์ของระบบที่พัฒนาขึ้น 51
รูปที่ 4.31 ข้อมูลดิบของจำนวนการโบกมือของผู้ใช้งานในการเข้าใช้แคนวาสครั้งก่อนหน้า บริเวณชั้น 12 อาคารเจริญวิศวกรรม 53
รูปที่ 4.32 ข้อมูลดิบของหมายเลขแคนวาสที่ถูกเรียกใช้งานบริเวณชั้น 12 อาคารเจริญวิศวกรรม 53
รูปที่ 4.33 ข้อมูลดิบของจำนวนครั้งที่ระบบถูกกระตุ้นให้เข้าสู่แคนวาสปริยายจากคนที่เดินผ่าน บริเวณชั้น 12 อาคารเจริญวิศวกรรม 54
รูปที่ 4.34 ข้อมูลดิบของจำนวนการโบกมือของผู้ใช้งานในการเข้าใช้แคนวาสครั้งก่อนหน้า บริเวณชั้น 13 อาคารเจริญวิศวกรรม 54
รูปที่ 4.35 ข้อมูลดิบของหมายเลขแคนวาสที่ถูกเรียกใช้งานบริเวณชั้น 13 อาคารเจริญวิศวกรรม 55
รูปที่ 4.36 ข้อมูลดิบของจำนวนครั้งที่ระบบถูกกระตุ้นให้เข้าสู่แคนวาสปริยายจากคนที่เดินผ่าน บริเวณชั้น 13 อาคารเจริญวิศวกรรม 55
รูปที่ 4.37 ข้อมูลดิบของจำนวนการโบกมือของผู้ใช้งานในการเข้าใช้แคนวาสครั้งก่อนหน้า บริเวณอาคารปฏิบัติการไฟฟ้าแรงสูง 56
รูปที่ 4.38 ข้อมูลดิบของหมายเลขแคนวาสที่ถูกเรียกใช้งานบริเวณอาคารปฏิบัติการไฟฟ้าแรงสูง 56
รูปที่ 4.39 ข้อมูลดิบของจำนวนครั้งที่ระบบถูกกระตุ้นให้เข้าสู่แคนวาสปริยายจากคนที่เดินผ่าน บริเวณอาคารปฏิบัติการไฟฟ้าแรงสูง 57
รูปที่ 4.40 ร้อยละของผู้ใช้งานที่มีการโต้ตอบกับระบบที่พัฒนาขึ้นในแต่ละจุดติดตั้ง 58
รูปที่ 4.41 ร้อยละของผู้ใช้งานที่โต้ตอบกับโปรแกรมในแต่ละแคนวาสของบริเวณชั้น 12 อาคารเจริญวิศวกรรม 61
รูปที่ 4.42 ร้อยละของผู้ใช้งานที่โต้ตอบกับโปรแกรมในแต่ละแคนวาสของบริเวณชั้น 13 อาคารเจริญวิศวกรรม 61
รูปที่ 4.43 ร้อยละของผู้ใช้งานที่โต้ตอบกับโปรแกรมในแต่ละแคนวาสของบริเวณอาคารปฏิบัติการไฟฟ้าแรงสูง 62

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การใช้พลังงานภายในประเทศไทยปัจจุบันมีแนวโน้มเพิ่มขึ้นอย่างต่อเนื่องทำให้ต้องผลิตไฟฟ้าเพื่อรองรับความต้องการใช้งานในอนาคต ส่งผลโดยตรงต่อปริมาณการปล่อยแก๊สเรือนกระจกสู่บรรยากาศ จากการศึกษาพบว่าภาคอาคารสำนักงานรวมถึงบ้านพักอาศัยมีการใช้พลังงานมากกว่าร้อยละ 20 ของการใช้พลังงานทั้งหมด [1] ดังนั้นการวางแผนลดการใช้พลังงานในอาคาร ถือเป็นแนวทางที่ช่วยลดการใช้พลังงานได้อย่างมีประสิทธิภาพ

ปัจจุบันเทคโนโลยีที่ถูกนำมาใช้ในระบบจัดการพลังงานมีอยู่อย่างหลากหลาย เช่น BACnet [2], LonWorks [3], Modbus [4] ซึ่งมีรูปแบบมาตรฐานการสื่อสารที่แตกต่างกัน ดังนั้นหากสามารถเลือกใช้งานเทคโนโลยีต่าง ๆ ที่มีอยู่เดิมนี้ร่วมกันในระบบจัดการพลังงานเดียวกันได้จะทำให้เกิดความยืดหยุ่นและประสิทธิภาพในการใช้งานมากยิ่งขึ้นด้วย ในวิทยานิพนธ์นี้ได้ศึกษามาตรฐานการสื่อสารที่มีชื่อว่า "IEEE1888" [5] หรือเรียกอีกชื่อหนึ่งว่า "โพรโทคอลโครงข่ายการควบคุมชุมชนสีเขียวอย่างแพร่หลาย" (ubiquitous green community control network protocol : UGCC) ซึ่งมีความสามารถเชื่อมโยงมาตรฐานการสื่อสารแบบต่าง ๆ ที่มีอยู่เดิมให้เป็นมาตรฐานเดียวกันกล่าวคือมาตรฐาน IEEE1888 สื่อสารกันด้วยภาษา xml ที่มีใช้อย่างทั่วไปด้วยลักษณะ URI รูปแบบเดียวกันผ่านโครงข่าย TCP/IP และด้วยความสามารถดังกล่าวนี้ได้มีการนำมาตรฐาน IEEE1888 มาประยุกต์ใช้งาน เช่น ในงานวิจัย [6] ใช้มาตรฐาน IEEE1888 ในระบบสื่อสารเพื่อตอบสนองอุปสงค์แบบอัตโนมัติสำหรับการบริหารจัดการในช่วงเวลาที่ความต้องการไฟฟ้าสูงสุด ในงานวิจัย [7] ได้นำมาตรฐาน IEEE1888 ไปใช้ร่วมกับเทคโนโลยีสารสนเทศและการสื่อสารรวมทั้งเทคโนโลยีอินเทอร์เน็ตอนาคตสำหรับจัดการพลังงานของโครงการมหาวิทยาลัยโตเกียวสีเขียว ในงานวิจัย [8] และ [9] นำมาตรฐาน IEEE1888 มาใช้เฝ้าสังเกตสภาพอากาศในระบบจัดการพลังงานในอาคารในภาควิชาวิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชาวิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัยมีการจัดตั้งโครงการเพื่อสร้างระบบจัดการพลังงานในอาคาร (CU-BEMS) ซึ่งในระบบประกอบไปด้วยส่วนประกอบต่าง ๆ รวมถึงตัวรับรู้ที่ถูกติดตั้งอยู่ภายในภาควิชา ทั้งสิ้น 200 จุด โดยในแต่ละจุดจะประกอบไปด้วย ตัวรับรู้หลายประเภท คือ ตัวรับรู้ความเข้มแสงสว่าง, ตัวตรวจจับการเคลื่อนไหว, ตัวรับรู้อุณหภูมิ, ตัวรับรู้ความชื้นสัมพัทธ์รวมถึงตัววัดกำลังงานไฟฟ้า ข้อมูลที่วัดได้จากตัวรับรู้เหล่านี้จะถูกส่งยังหน่วยเก็บข้อมูลอย่างสม่ำเสมอ ดังนั้นจำเป็นต้องมีการสร้างโปรแกรมประยุกต์ในการนำเสนอข้อมูลดิบเหล่านี้ในรูปแบบต่าง ๆ หรือเรียกว่าการสร้างมโนภาพ (visualization) ให้กับข้อมูลเพื่อทำให้ผู้ใช้สามารถเข้าถึงข้อมูลได้โดยง่ายและทราบถึงข้อมูลการใช้พลังงานเพื่อให้สามารถวางแผนบริหารการใช้งานพลังงานให้เป็นไปอย่างมีประสิทธิภาพมากยิ่งขึ้น

1.2 วัตถุประสงค์ของงานวิทยานิพนธ์

1. พัฒนาโปรแกรมสร้างมโนภาพของข้อมูลการใช้พลังงานในระบบจัดการพลังงานในอาคารบริเวณภาควิชาวิศวกรรมไฟฟ้า บนพื้นฐาน IEEE1888 โดยใช้โปรแกรม processing แสดงผลการนำเสนอข้อมูลโดยใช้หน่วยแสดงผลภายนอก และโต้ตอบกับโปรแกรมด้วยตัวรับรู้ท่าทางในการรับข้อมูลจากผู้ใช้ เช่น การยกมือ, การโบกมือ เป็นต้น
2. ประเมินโปรแกรมโดยสร้างแบบสอบถามให้ผู้ใช้งานตอบหลังจากใช้งานโปรแกรม และเขียนโปรแกรมให้เก็บข้อมูลพฤติกรรมการใช้งานของผู้ใช้งานเพื่อส่งไปเก็บไว้ในหน่วยเก็บข้อมูลบนพื้นฐาน IEEE1888

1.3 ขอบเขตวิทยานิพนธ์

1. ใช้โปรแกรม processing ในการออกแบบโปรแกรมสร้างมโนภาพเชิงโต้ตอบของข้อมูลการใช้พลังงานในระบบจัดการพลังงานในอาคารบริเวณภาควิชาวิศวกรรมไฟฟ้าของโครงการ CU-BEMS
2. การสื่อสารระหว่างโปรแกรมสร้างมโนภาพกับหน่วยเก็บข้อมูลของมาตรฐาน IEEE1888 ใช้เพียงโพรโทคอล FETCH และ WRITE เท่านั้น
3. แสดงผลการนำเสนอข้อมูลโดยใช้หน่วยหน่วยแสดงผลภายนอกเท่านั้น
4. การโต้ตอบระหว่างผู้ใช้งานกับโปรแกรม ใช้การยกมือ และการโบกมือ
5. ประเมินผลโปรแกรมโดยใช้แบบสอบถามหลังจากใช้งานโปรแกรม และเขียนโปรแกรมให้เก็บข้อมูลพฤติกรรมการใช้งานของผู้ใช้งานเพื่อส่งไปเก็บไว้ในหน่วยเก็บข้อมูลบนพื้นฐาน IEEE1888

1.4 ขั้นตอนและวิธีการดำเนินงาน

1. ศึกษาบทความ วิทยานิพนธ์ และงานวิจัยในอดีตที่เกี่ยวข้องกับการสร้างโปรแกรมประยุกต์สำหรับการเฝ้าสังเกตุในระบบจัดการพลังงานภายในอาคาร
2. ศึกษาเกี่ยวกับมาตรฐานการสื่อสารใน IEEE1888
3. ศึกษาเกี่ยวกับคุณลักษณะต่าง ๆของโปรแกรม processing และตัวรับรู้การแสดงท่าทาง
4. ออกแบบและสร้างโปรแกรมประยุกต์ที่มีการสร้างมโนภาพในเชิงโต้ตอบโดยอาศัยตัวรับรู้การแสดงท่าทางเพื่อการเฝ้าสังเกตุในระบบจัดการพลังงานภายในอาคาร
5. วิเคราะห์และประเมินผลประสิทธิภาพของโปรแกรมประยุกต์
6. สรุปและเผยแพร่ผลการวิจัยลงในบทความ และจัดทำเล่มวิทยานิพนธ์ฉบับสมบูรณ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

องค์ความรู้ในการพัฒนาโปรแกรมสร้างมโนภาพเชิงโต้ตอบโดยใช้ตัวรับรู้ท่าทางเพื่อรับข้อมูลอินพุตจากผู้ใช้งานในการควบคุมการทำงานของโปรแกรมสำหรับการใช้งานร่วมกับระบบจัดการพลังงานในอาคารบนพื้นฐานของมาตรฐาน IEEE1888

1.6 ประมวลวิทยานิพนธ์

บทที่ 1 บทนำ: กล่าวถึงความเป็นมา และความสำคัญของปัญหา วัตถุประสงค์ของการทำวิทยานิพนธ์ แนวทางและขอบเขตของวิทยานิพนธ์ ขั้นตอนการดำเนินงาน รวมถึงประโยชน์ที่ได้รับ

บทที่ 2 หลักการและงานวิจัยในอดีตที่เกี่ยวข้อง: กล่าวถึงงานวิจัยในอดีตที่เกี่ยวของกับการสร้างมโนภาพให้กับข้อมูล และกล่าวถึงหลักการพื้นฐานต่างๆที่นำมาใช้ในวิทยานิพนธ์ฉบับนี้ ซึ่งประกอบด้วย มาตรฐาน IEEE1888 การสร้างมโนภาพข้อมูลด้วยภาษา processing รวมทั้งกล่าวถึงตัวรับรู้ท่าทาง (gesture sensor) ที่นำมาใช้ในวิทยานิพนธ์ฉบับนี้

บทที่ 3 สถาปัตยกรรมระบบ และการทดสอบเบื้องต้น: กล่าวถึงภาพรวมของระบบที่พัมนาขึ้น การทดสอบการทำงานร่วมกันของแต่ละองค์ประกอบนอกจากนั้นกล่าวถึงการสร้างโปรแกรมต้นแบบและการประเมินโปรแกรมต้นแบบเพื่อใช้ในการพัฒนาก่อนการบูรณาการเข้ากับระบบที่พัฒนาขึ้นเพื่อใช้งานจริง

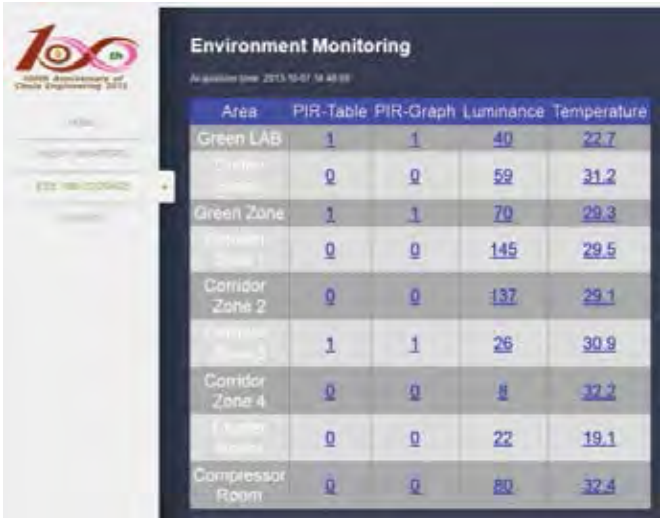
บทที่ 4 การพัฒนาโปรแกรมสร้างมโนภาพเชิงโต้ตอบด้วยตัวรับรู้ท่าทางตามมาตรฐาน IEEE1888 เพื่อการใช้งานจริงในโครงการ CU-BEMS: กล่าวถึงการออกแบบโปรแกรมสร้างมโนภาพทั้งในส่วนของการแสดงผลหลัก และข้อมูลการตั้งค่าต่าง ๆ ภายในโปรแกรม รวมทั้งกล่าวถึงการทดสอบเพื่อประเมินพฤติกรรมผู้เข้าใช้งานระบบโดยแบ่งเป็นกรณีศึกษาต่าง ๆ

บทที่ 5 บทสรุปและข้อเสนอแนะ: สรุปงานวิจัยทั้งหมดในวิทยานิพนธ์ฉบับนี้และเสนอแนวทางในการพัฒนางานวิจัยต่อไป

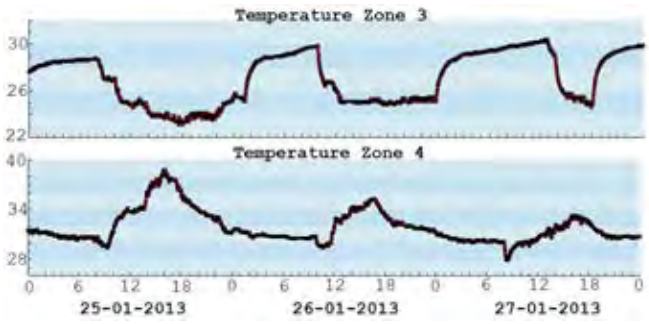
บทที่ 2

หลักการและงานวิจัยในอดีตที่เกี่ยวข้อง

จากการศึกษางานวิจัยในอดีตที่เกี่ยวข้องกับระบบจัดการพลังงานในอาคารรวมทั้งในกริดอัจฉริยะนั้นมีการสร้างมโนภาพของข้อมูลในลักษณะต่าง ๆ เช่น โครงการมหาวิทยาลัยโตเกียวสีเขียว [7] แสดงข้อมูลการใช้พลังงานในอาคารโดยรวมแยกตามอาคารเป็นกราฟแท่งบนอนุกรมเวลา งานวิจัย [8] และ [9] เลือกรูปแบบการนำเสนอข้อมูลสภาพอากาศในระบบจัดการพลังงานในอาคารโดยสร้างโปรแกรมประยุกต์บนเว็บเพื่อใช้แสดงผลดังแสดงในรูปที่ 2.1(a) และใช้กราฟเส้นบนอนุกรมเวลาเพื่อแสดงแนวโน้มของสภาพอากาศดังแสดงในรูปที่ 2.1(b)



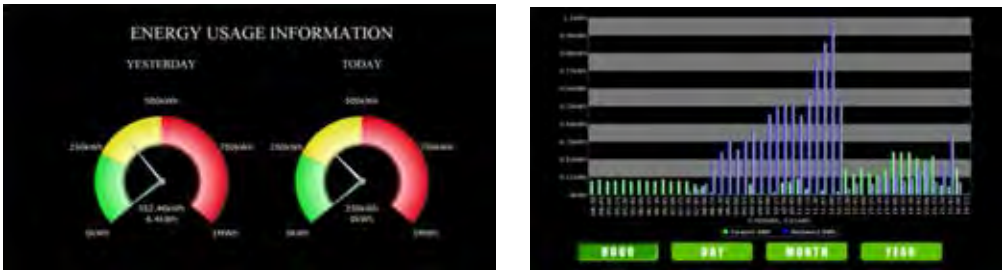
(a) ส่วนติดต่อผู้ใช้



(b) การแสดงผลข้อมูลโดยใช้กราฟเส้นบนอนุกรมเวลา

รูปที่ 2.1: โปรแกรมประยุกต์บนเว็บเพื่อเฝ้าสังเกตสภาพอากาศในอาคารบริเวณภาควิชาวิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัย ในงานวิจัย [8], [9]

งานวิจัย [10] เป็นการใช้ระบบทดสอบกริดอัจฉริยะในอาคารขนาดเล็กขนาด 2 ชั้น ซึ่งตั้งอยู่บนเกาะเจจู ประเทศเกาหลี เพื่อเฝ้าสังเกตุการใช้พลังงานในอาคาร และการผลิตกระแสไฟฟ้าจากแหล่งพลังงานทดแทน ซึ่งข้อมูลดังกล่าวถูกนำมาแสดงบนแผงหน้าปัดหลักเพื่อเปรียบเทียบการใช้พลังงานระหว่างปัจจุบันกับวันก่อนหน้าดังรูปที่ 2.2(a) ส่วนในรูปที่ 2.2(b) แสดงการใช้พลังงานในแต่ละช่วงเวลา ซึ่งแสดงเป็นกราฟแท่งบนอนุกรมเวลา



(a) แผงหน้าปัดหลัก (b) การแสดงผลข้อมูลตามช่วงเวลา

รูปที่ 2.2: ข้อมูลการใช้พลังงานของระบบทดสอบกริดอัจฉริยะในอาคารขนาดเล็กบนเกาะเจจู ประเทศเกาหลีใต้ [10]

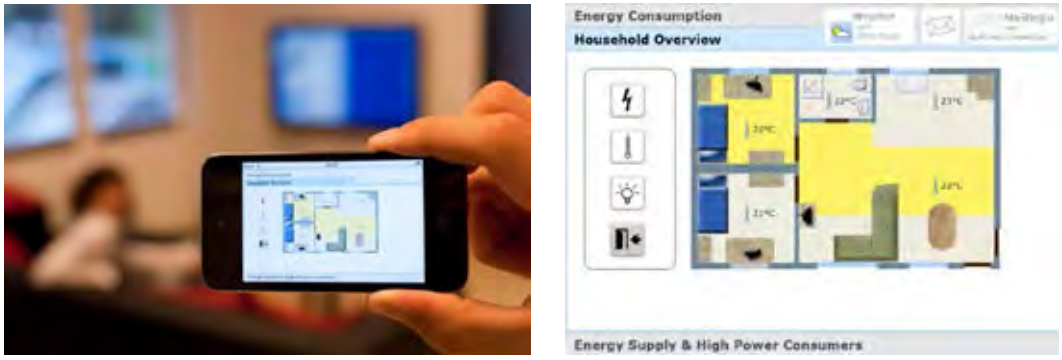
นอกจากนั้นยังมีส่วนต่อประสานกับผู้ใช้ (user interface) ซึ่งแสดงการใช้ไฟฟ้าโดยรวมของระบบ และแสดงการใช้ไฟฟ้าของหลอดไฟ รถยนต์ไฟฟ้า แบตเตอรี่ แหล่งกำเนิดพลังงานทดแทนโดยแสดงผลแยกส่วนกันดังแสดงในรูปที่ 2.3

รูปที่ 2.3: ส่วนต่อประสานกับผู้ใช้ของะบบทดสอบกริดอัจฉริยะในอาคารขนาดเล็กบนเกาะเจจู ประเทศเกาหลีใต้ [10]

งานวิจัย [11] ได้ประยุกต์ใช้งานวิจัยที่ [10] เพื่อเฝ้าสังเกตการใช้ไฟฟ้าของห้องชุด Mudeung Park จำนวน 20 ห้อง ซึ่งตั้งอยู่ในบริเวณเมืองที่มีการอยู่อาศัยหนาแน่น โดยมีการติดตั้งหน่วยแสดงผลไว้ทุกห้องซึ่งมีรูปแบบของส่วนต่อประสานกับผู้ใช้ตามรูปที่ 2.4 แสดงการใช้พลังงานโดยรวม และมีการแยกแสดงการใช้พลังงานสำหรับเครื่องใช้ไฟฟ้าแต่ละชนิด

รูปที่ 2.4: ส่วนต่อประสานกับผู้ใช้ในอาคารชุด Mudeung Park [11]

งานวิจัย [12] นำเสนอส่วนต่อประสานกับผู้ใช้สำหรับบ้านอัจฉริยะของโครงการ MeRegioMobil ในประเทศเยอรมัน โดยส่วนต่อประสานกับผู้ใช้ของระบบจัดการพลังงานในระบบบ้านอัตโนมัตินี้ถูกเรียกว่า แผงจัดการพลังงาน (Energy Management Panel : EMP) ซึ่งมีทั้งบนโทรศัพท์เคลื่อนที่ และบนเว็บดังแสดงในรูปที่ 2.5 วัตถุประสงค์ของแผงจัดการพลังงานเพื่อให้ผู้ใช้ทราบถึงข้อมูลการใช้งานไฟฟ้าในบ้าน และกำหนดการใช้งานในระบบจัดการพลังงาน

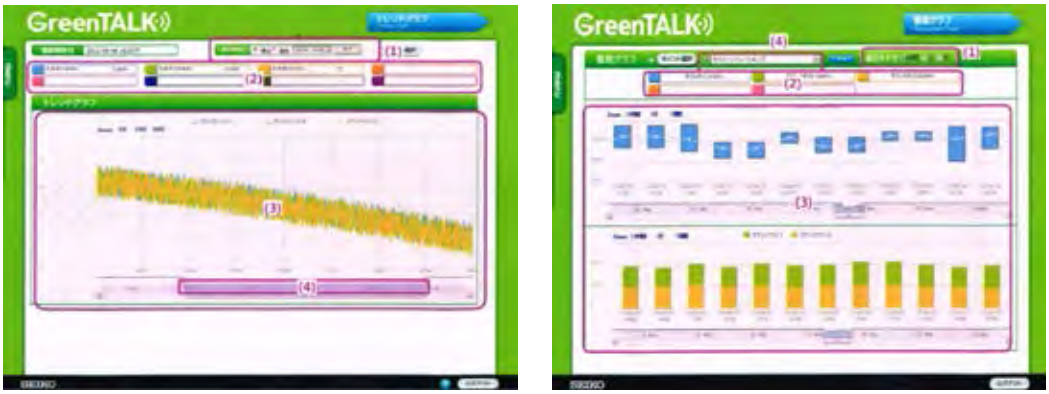


(a) แผงจัดการพลังงานบนโทรศัพท์มือถือ (b) แผงจัดการพลังงานภายในบ้าน

รูปที่ 2.5: แผงจัดการพลังงานสำหรับบ้านอัจฉริยะของโครงการ MeRegioMobil [12]

สำหรับโปรแกรมประยุกต์เพื่อเฝ้าสังเกตุการใช้ไฟฟ้ารวมถึงสภาพอากาศในโครงการ CU-BEMS มี 2 โปรแกรม ซึ่งได้รับความร่วมมือกับ บริษัท เอ็นดีอาร์ โซลูชั่น (ประเทศไทย) จำกัด ในการพัฒนาโปรแกรมประยุกต์ CU-BEMS และบริษัท ไซโก พรีซิชั่น (ประเทศไทย) จำกัด ในการพัฒนาโปรแกรมประยุกต์ Greentalk โดยรูปแบบการนำเสนอข้อมูลของโปรแกรมประยุกต์ Greentalk ใช้

กราฟเส้น และแผนภูมิแท่งเป็นหลักดังแสดงในรูปที่ 2.6(a) และรูปที่ 2.6(b) ตามลำดับ ซึ่งผู้ใช้สามารถเลือกแสดงผลข้อมูลในตำแหน่ง และช่วงเวลาที่สนใจ



(a) การแสดงผลข้อมูลโดยกราฟเส้น (b) การแสงผลข้อมูลโดยใช้แผนภูมิแท่ง

รูปที่ 2.6: โปรแกรมประยุกต์ Greentalk

โปรแกรมประยุกต์ CU-BEMS มีรูปแบบการนำเสนอข้อมูลโดยการใช้กราฟเส้น แผนภูมิแท่ง แผนภูมิวงกลม ร่วมกับการใช้กราฟฟิกเพื่อดึงดูดความสนใจของผู้ใช้ และมีการแสดงผลแยกตามบริเวณที่ผู้ใช้งานสนใจ เช่น ข้อมูลการใช้พลังงานรวมถึงสภาพแวดล้อมโดยรวมของอาคารแยกตามชั้น และตามโซน โดยรูปที่ 2.7 แสดงส่วนต่อประสานหลักของโปรแกรม CU-BEMS มีการใช้สีเพื่อบ่งบอกถึงการใช้ไฟฟ้าในแต่ละตึก และรูปที่ 2.8 แสดงรูปแบบของโปรแกรมประยุกต์ CU-BEMS เมื่อผู้ใช้งานสนใจเลือกดูข้อมูลแยกตามบริเวณ โดยรูปที่ 2.8(a) แสดงข้อมูลในอาคารที่ผู้ใช้สนใจ และรูปที่ 2.8(b) แสดงข้อมูลในชั้นที่ผู้ใช้สนใจ

รูปที่ 2.7: ส่วนต่อประสานหลักของโปรแกรม CU-BEMS

(a) การแสดงผลข้อมูลแยกตามอาคารที่ผู้ใช้สนใจ

(b) การแสงผลข้อมูลแยกตามอาคารที่ผู้ใช้สนใจ

รูปที่ 2.8: โปรแกรมประยุกต์ CU-BEMS

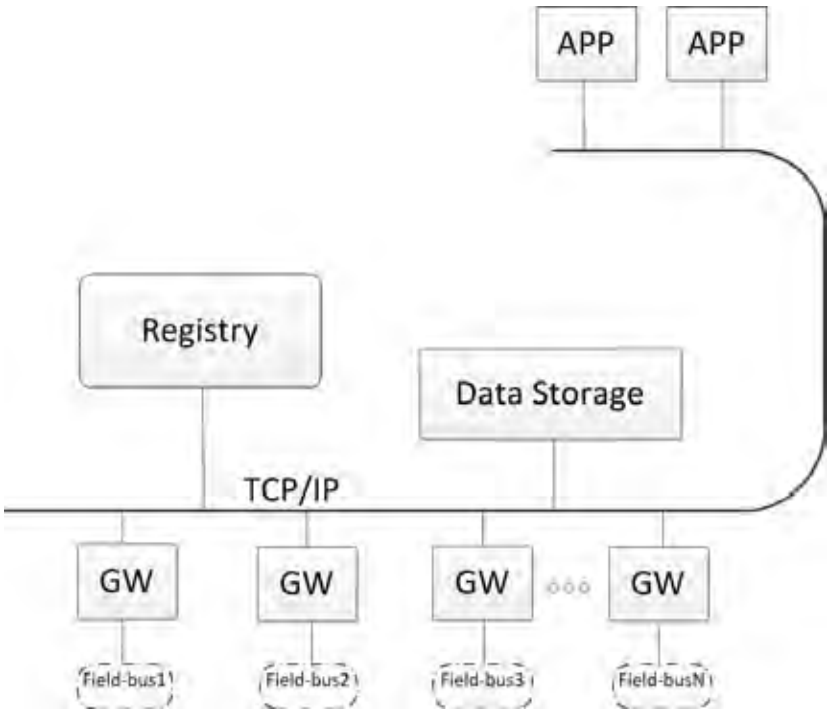
นอกจากรูปแบบของการสร้างมโนภาพให้กับข้อมูลแล้วสิ่งที่จะช่วยทำให้การนำเสนอข้อมูลเป็นที่น่าสนใจ และสามารถกระตุ้นให้เกิดการตระหนักรู้ถึงการใช้พลังงานมากยิ่งขึ้น คือ การเพิ่มการโต้ตอบระหว่างผู้ใช้งานกับโปรแกรมประยุกต์ โดยรูปแบบการโต้ตอบระหว่างผู้ใช้งานกับโปรแกรมประยุกต์ในแบบดั้งเดิมที่ใช้กันอย่างแพร่หลาย เช่น การใช้เมาส์ (mouse) และแผงแป้นอักขระ (keyboard) วิทยานิพนธ์ฉบับนี้ขอนำเสนอโปรแกรมที่มีรูปแบบการโต้ตอบกับโปรแกรมโดยใช้การแสดงท่าทาง เช่น ใช้การยกมือขึ้น, ใช้การโบกมือ ซึ่งการแสดงท่าทางเป็นการสื่อสารตามธรรมชาติซึ่งคนทั่วไปคุ้นเคย [21] และใช้การนำเสนอข้อมูลในเชิงทัศนศิลป์ (visual art) ซึ่งผู้ใช้ไม่จำเป็นต้องมีความสามารถในการใช้งานคอมพิวเตอร์ และไม่จำเป็นต้องใช้การคิดวิเคราะห์มาก ทำให้ผู้ใช้งานทั่วไปสามารถใช้งานโปรแกรมประยุกต์ได้ง่าย และตระหนักรู้ถึงข้อมูลการใช้พลังงานรวมถึงข้อมูลสภาพแวดล้อมมากขึ้น มีส่วนทำให้ผู้ใช้งานช่วยกันลดการใช้พลังงานในช่วงเวลาที่มีความจำเป็นทำให้การใช้งานพลังงานไฟฟ้าในระบบเป็นไปอย่างมีประสิทธิภาพ

2.1 มาตรฐาน IEEE1888

มาตรฐาน IEEE1888 คือ มาตรฐานการสื่อสารที่เกี่ยวข้องกับการจัดการพลังงานซึ่งได้รับการอนุมัติจากสถาบันวิชาชีพวิศวกรไฟฟ้าและอิเล็กทรอนิกส์ในปีคริสต์ศักราช 2011 มีวัตถุประสงค์เพื่อใช้เชื่อมโยงระหว่างมาตรฐานในอุปกรณ์ และระบบของผู้ผลิตรวมถึงผู้ให้บริการรายต่าง ๆ ซึ่งมีการสื่อสารในรูปแบบที่แตกต่างกันให้สามารถทำงานบนมาตรฐานเดียวกันเพื่อใช้ในระบบจัดการพลังงานในอาคาร

2.1.1 สถาปัตยกรรมของ IEEE1888

สถาปัตยกรรมของมาตรฐาน IEEE1888 ประกอบไปด้วยส่วนประกอบหลักตามรูปที่ 2.9 แบ่งเป็นส่วนที่ทำหน้าที่ในระนาบข้อมูล คือ เกตเวย์ (GW), หน่วยเก็บข้อมูล (data storage) และโปรแกรมประยุกต์ (APP) ซึ่งถูกเรียกว่าองค์ประกอบ (component) อีกส่วนหนึ่งคือส่วนที่ทำหน้าที่ในระนาบควบคุม คือ รีจีสทรี (registry) ซึ่งทั้งสองส่วนนี้สื่อสารกันโดยใช้มาตรฐาน TCP/IP



รูปที่ 2.9: สถาปัตยกรรมของมาตรฐาน IEEE1888 [5]

- **เกตเวย์**
  ทำหน้าที่ในการเชื่อมต่อระหว่างเกตเวย์ (ตัวรับรู้และตัวกระตุ้น), หน่วยเก็บข้อมูลและโปรแกรมประยุกต์ ให้รูปแบบการสื่อสารเป็นไปตามมาตรฐาน IEEE1888

- **หน่วยเก็บข้อมูล**
  ทำหน้าที่ในการเก็บข้อมูลที่วัดได้จากตัวรับรู้เรียงตามเวลา

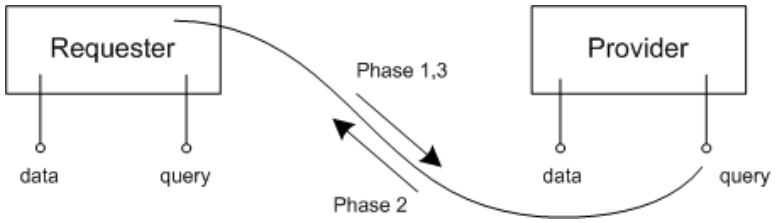
- **โปรแกรมประยุกต์**
  ทำหน้าที่เป็นส่วนติดต่อผู้ใช้งานในการเฝ้าสังเกตข้อมูลที่วัดได้จากตัวรับรู้เพื่อสามารถวางแผนการใช้ไฟฟ้า

- **รีจิสทรี**
  ทำหน้าที่เป็นระบบควบคุมการทำงานระหว่าง เกตเวย์ หน่วยเก็บข้อมูลและโปรแกรมประยุกต์

2.1.2 การสื่อสารพื้นฐานในมาตรฐาน IEEE1888

โพรโทคอลที่ถูกใช้ในการสื่อสารระหว่างองค์ประกอบ (component-to-component communication protocol) ถูกแบ่งออกเป็น 3 ชนิดคือ

1. โพรโทคอล FETCH (FETCH protocol) ใช้ในการค้นคืนข้อมูลจากองค์ประกอบระยะไกล มีองค์ประกอบที่เกี่ยวข้องตามรูปที่ 2.10 ซึ่งแสดงการรับส่งข้อมูลระหว่างผู้ร้องขอข้อมูล (requester) และผู้ให้บริการข้อมูล (provider)



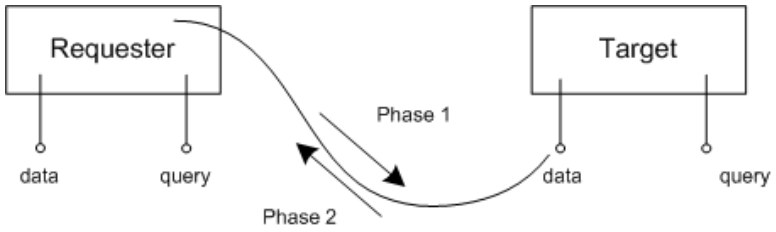
รูปที่ 2.10: โพลโทคอล FETCH [5]

เฟส 1 : ผู้ร้องขอข้อมูลส่งคำสั่งร้องขอข้อมูลไปยังผู้ให้บริการข้อมูลโดยผู้ร้องขอข้อมูลควรระบุช่วงเวลาของข้อมูลที่ต้องการรวมถึงขนาดสูงสุดของข้อมูลที่สามารถรับได้

เฟส 2 : ผู้ให้บริการข้อมูลส่งข้อมูลที่ถูกแบ่งออกเป็นชุดย่อย ๆ กลับมายังผู้ร้องขอข้อมูลโดยหากข้อมูลที่ร้องขอนั้นมีขนาดใหญ่กว่าขนาดสูงสุดของข้อมูลที่สามารถรับได้ ชุดข้อมูลจะถูกส่งมาพร้อมกับตัวชี้ตำแหน่งของข้อมูล (cursor) ซึ่งเป็นการแจ้งว่ายังมีข้อมูลชุดถัดไปซึ่งจะรอทำการส่งจนกว่าจะถึงข้อมูลชุดสุดท้ายที่ไม่มีตัวชี้ตำแหน่งของข้อมูลจึงสิ้นสุดการทำงานของโพรโทคอล FETCH

เฟส 3 : หากยังมีตัวชี้ตำแหน่งของข้อมูลในชุดข้อมูลที่ได้รับจากเฟสที่ 2 ผู้ร้องขอข้อมูลจะทำการส่งคำสั่งร้องขอข้อมูลพร้อมกับตัวชี้ตำแหน่งข้อมูลที่ได้รับไปยังผู้ให้บริการข้อมูล และกลับไปเริ่มทำงานในเฟสที่ 2 อีกครั้ง

2. โพรโทคอล WRITE (WRITE protocol) ใช้ในการถ่ายโอนข้อมูลไปยังองค์ประกอบระยะไกล มีองค์ประกอบที่เกี่ยวข้องตามรูปที่ 2.11 ซึ่งแสดงการรับส่งข้อมูลระหว่างผู้ร้องขอในการส่งข้อมูล และเป้าหมาย (target)

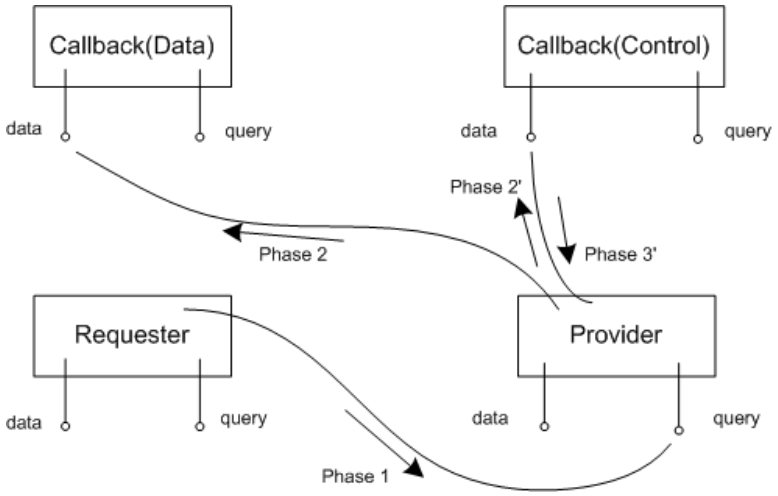


รูปที่ 2.11: โพลโทคอล WRITE [5]

เฟส 1 : ผู้ร้องขอในการส่งข้อมูลส่งข้อมูลพร้อมคำสั่งเขียนข้อมูลไปยังเป้าหมาย

เฟส 2 : เป้าหมายตอบกลับไปยังผู้ร้องขอในการส่งข้อมูลว่าข้อมูลที่ได้รับมาถูกต้องหรือเกิดข้อผิดพลาดขึ้น

3. โพรโทคอล TRAP (TRAP protocol) ใช้เพื่อจดทะเบียนเหตุการณ์ที่ต้องการให้แจ้งเตือนและทำการถ่ายโอนข้อมูลเมื่อเกิดเหตุการณ์ที่ทำการจดทะเบียนไว้ มีองค์ประกอบที่เกี่ยวข้องตามรูปที่ 2.12 ซึ่งแสดงการรับส่งข้อมูลระหว่าง ผู้ร้องขอข้อมูล, ผู้ให้บริการข้อมูล, ผู้เรียกกลับของข้อมูล และผู้เรียกกลับของส่วนควบคุม



รูปที่ 2.12: โพรโทคอล TRAP [5]

เฟส 1 : ผู้ร้องขอข้อมูลทำการจดทะเบียนเหตุการณ์ที่ต้องการให้ผู้ให้บริการข้อมูลแจ้งเตือนโดยส่งคำสั่งร้องขอข้อมูลไปยังผู้ให้บริการข้อมูล ซึ่งต้องระบุเวลาหมดอายุของคำสั่งในหน่วยวินาทีรวมทั้งระบุ URI สำหรับใช้การตอบกลับทั้งของผู้เรียกกลับของข้อมูล และ ผู้เรียกกลับของส่วนควบคุม

เฟส 2 : ผู้ให้บริการข้อมูลตอบกลับไปยัง URI ของผู้เรียกกลับของข้อมูลเมื่อเกิดเหตุการณ์ที่ตรงกับผู้ร้องขอข้อมูลได้ทำการจดทะเบียนไว้

เฟส 3 : ผู้เรียกกลับของข้อมูลตอบกลับไปยังผู้ให้บริการข้อมูลว่าข้อมูลที่ได้รับมาถูกต้องหรือเกิดข้อผิดพลาดขึ้น

เฟส 2' : ผู้ให้บริการข้อมูลจะประกาศข้อผิดพลาดของการสื่อสารไปยัง URI ของผู้เรียกกลับส่วนควบคุมเมื่อเกิดข้อผิดพลาดขึ้นในเฟสที่ 3

เฟส 3' : ผู้เรียกกลับส่วนควบคุมตอบกลับไปยังผู้ให้บริการข้อมูลว่าข้อมูลที่ได้รับมาถูกต้องหรือเกิดข้อผิดพลาดขึ้น

2.1.3 การประยุกต์ใช้มาตรฐาน IEEE1888

โครงการ CU-BEMS มีการใช้องค์ประกอบหลักเช่นเดียวกับในมาตรฐาน IEEE1888 ได้แก่ ตัวรับรู้ในการวัดสภาพแวดล้อม ซึ่งในโครงการประกอบด้วย ตัวรับรู้อุณหภูมิ, ความชื้นสัมพัทธ์, ความเข้มแสงสว่าง และตัวตราจจับการเคลื่อนไหว ดังรูปที่ 2.13(a) และเกตเวย์ทำหน้าที่ส่งข้อมูลให้ตรงตามมาตรฐาน IEEE1888 ดังรูปที่ 2.13(b)

(a) ตัวรับรู้สภาพแวดล้อม (b) เกตเวย์

รูปที่ 2.13: ตัวรับรู้ และเกตเวย์ที่ใช้โครงการ CU-BEMS

ข้อมูลที่ถูกส่งมาจากเกตเวย์ถูกเก็บไว้ที่หน่วยเก็บข้อมูลดังรูปที่ 2.14 ซึ่งติดตั้งไว้ที่ห้องเครือข่ายท้องถิ่นภายในห้องปฏิบัติการวิจัยระบบโทรคมนาคม ชั้น 13 อาคารเจริญวิศวกรรม

รูปที่ 2.14: หน่วยเก็บข้อมูลที่ใช้โครงการ CU-BEMS

วิทยานิพนธ์นี้ใช้ประโยชน์จากโพรโทคอล FETCH ในการดึงข้อมูลการใช้ไฟฟ้ารวมถึงข้อมูลสภาพแวดล้อมจากหน่วยเก็บข้อมูลของโครงการ CU-BEMS มาใช้ในการแสดงผลแสดงผล และใช้

โพรโตอคอล WRITE ในการเก็บข้อมูลจำนวนผู้เข้าใช้งานโปรแกรมรวมถึงระยะเวลาในการใช้งานของผู้ใช้แต่ละคน ในส่วนของโพรโทคอล TRAP ไม่ถูกนำมาใช้ในงานวิจัยนี้

2.2 การสร้างมโนภาพข้อมูลด้วยภาษา processing

การสร้างมโนภาพข้อมูล คือ เทคนิคในการสร้างภาพ แผนผัง หรือภาพเคลื่อนไหว เพื่อแปลงข้อมูลที่เป็นตัวเลข และตัวอักษรซึ่งเป็นข้อมูลดิบให้เป็นภาพที่สามารถแสดงรูปทรง หรือแสดงความสัมพันธ์ระหว่างข้อมูลทำให้สามารถมองเห็นการเปลี่ยนแปลง และทำความเข้าใจได้ง่ายขึ้น [13]

ขั้นตอนพื้นฐานในการสร้างมโนภาพให้กับข้อมูลสามารถแบ่งออกเป็น 7 ขั้นตอนดังนี้

1. การแสวงหาข้อมูล (acquiring data) : เป็นขั้นตอนการได้มาซึ่งข้อมูล
2. การแยกวิเคราะห์ข้อมูล (parsing data) : หลังจากขั้นตอนการแสวงหาข้อมูลจำเป้นต้องมีการแยกวิเคราะห์ข้อมูล เพื่อเปลี่ยนเป็นรูปแบบที่ตั้งใจจะใช้งาน
3. การกรองข้อมูล (filtering data) : เป็นขั้นตอนสำหรับกรองข้อมูลที่ไม่เกี่ยวข้องกับการใช้งาน
4. การทำเหมืองข้อมูล (mining data) : ขั้นตอนนี้จะเกี่ยวข้องกับคณิตศาสตร์สถิติ เช่น การคำนวณค่าสูงสุด และต่ำสุด เพื่อการนำเสนอข้อมูลบนหน้าจอในสเกลที่เหมาะสม
5. การแสดงข้อมูล (representing data) : เป็นขั้นตอนในการกำหนดรูปแบบพื้นฐานในการนำเสนอข้อมูลเพื่อสร้างมโนภาพซึ่งในแต่ละชุดข้อมูลอาจมีวิธีการนำเสนอไม่เหมือนกัน เช่น กราฟ, แถบรายการ, แผนภูมิต้นไม้ เป็นต้น
6. การปรับแต่ง (refining) : เป็นขั้นตอนการปรับแต่งรูปแบบการนำเสนอพื้นฐานให้ชัดเจน และน่าสนใจมากขึ้น
7. การปฏิสัมพันธ์ (interaction) : เป็นขั้นตอนในการเพิ่มการปฏิสัมพันธ์ โดยการเพิ่มวิธีในการจัดการกับข้อมูล หรือ เพิ่มการควบคุมคุณสมบัติต่าง ๆ

ในงานวิจัยนี้ได้ประยุกต์ใช้ขั้นตอนการสร้างมโนภาพข้อมูลพื้นฐานทั้ง 7 เพื่อพัฒนาโปรแกรมสร้างมโนภาพ ซึ่งการแสวงหาข้อมูลนั้นใช้ปรแกรม processing ในการ FETCH เพื่อดึงข้อมูลจากหน่วยเก็บข้อมูลของ CU-BEMS หลังจากนั้นเขียนโปรแกรม processing เพื่อใช้แยกวิเคราะห์, กรอง, และทำเหมืองข้อมูล ซึ่งในวิทยานิพนธ์นี้จะขอรวมเรียกขั้นตอนนี้ว่าการประมวลข้อมูลเตรียมให้อยู่ในรูปแบบที่สามารถใช้งานได้ง่าย และใช้หน่วยแสดงผลภายนอกสำหรับการแสดงข้อมูล สำหรับการปฏิสัมพันกับโปรแกรมที่พัฒนานั้นใช้ตัวรับรู้ท่าทางเพื่อรับอินพุตจากผู้ใช้เพื่อใช้งานระบบ สุดท้ายการปรับแต่งใช้เพื่อพัฒนาโปรแกรมสร้างมโนภาพให้มีปประสิทธิภาพมากยิ่งขึ้น

2.2.1 โปรแกรม processing

ภาษาโปรแกรม processing ถูกคิดค้นขึ้นใน ค.ศ. 2001 โดย Benjamin Fry และ Casey Reas จากสถาบันเทคโนโลยีแมสซาชูเซตส์ (MIT) ใช้พื้นฐานจากภาษาจาวา มีคุณสมบัติที่มุ่งเน้นทางด้านกราฟฟิกและสื่อประสม สามารถพัฒนาโปรแกรม และเชื่อมต่อเพื่อบูรณาการเข้ากับอุปกรณ์อื่น ๆ ได้ง่าย มีส่วนต่อประสานหลักของโปรแกรมดังแสดงในรูปที่ 2.15

รูปที่ 2.15: ส่วนต่อประสานหลักของโปรแกรม processing

2.2.2 การประยุกต์ใช้โปรแกรม processing

การสร้างมโนภาพให้กับข้อมูลจำเป็นต้องอาศัยเครื่องมือที่มีประสิทธิภาพเพื่อช่วยจัดการกับข้อมูลและบูรณาการเข้ากับอุปกรณ์อื่น ๆ ซึ่งในปัจจุบันมีเครื่องมือที่ใช้สร้างมโนภาพให้กับข้อมูลที่ได้รับความนิยมอย่างแพร่หลาย เช่น D3 [15], R [16], processing [17], gephi [18] เป็นต้น โดยวิทยานิพนธ์นี้เลือกใช้โปรแกรม processing ในการออกแบบและพัฒนาโปรแกรมสร้างมโนภาพ เนื่องจากเป็นโปรแกรมที่มีความสามารถในด้านการจัดการกับข้อมูลเพื่อออกแบบและพัฒนาในเชิงทัศนศิลป์ [13], [14] นอกจากนั้นยังเป็นโปรแกรมที่ง่ายต่อการบูรณาการเข้ากับอุปกรณ์หรือระบบอื่น ๆ เช่นมีความสามารถในการเชื่อมต่อเพื่อใช้งานร่วมกับไมโครคอมโทรเลอร์อาร์ดูอิโน [19] และจากการศึกษางานวิจัยในอดีตมีการนำโปรแกรม processing มาใช้เพื่อบูรณาการร่วมกับตัวรับรู้ท่าทางในงานแสดงศิลปะในเชิงเล่าเรื่องซึ่งมีชื่อว่า magic monkey ให้ผู้เข้าชมงานใช้การแสดงท่าทางเพื่อควบคุมตัวละคร และมีรูปแบบการนำเสนอโดยการเล่าเรื่องราวในสถานการณ์ต่าง ๆ เพื่อให้ผู้เข้าชมได้มีส่วนร่วมกับงานศิลปะมากขึ้น [20] สำหรับงานนี้ได้ข้อสรุปว่าผู้เข้าชมงานสามารถรับรู้เรื่องราวที่ผู้จัดแสดงงานศิลปะต้องการนำเสนอ และอยากมีส่วนร่วมกับการตอบโต้กับงานศิลปะที่แสดงมากขึ้น จากงานวิจัยนี้แสดงให้เห็นว่าโปรแกรม processing สามารถที่จะนำไปใช้ในการดึงดูดความสนใจของผู้เข้าร่วมงานได้ ผู้วิจัยจึงสนใจเลือกศึกษาการใช้โปรแกรม processing ในการนำเสนอข้อมูล เพื่อดึงดูดความสนใจของผู้ใช้งานโปรแกรมสร้างมโนภาพสำหรับระบบจัดการพลังงาน CU-BEMS

2.3 ตัวรับรู้ท่าทาง (gesture sensor)

งานวิจัยนี้ใช้ kinect จากบริษัทไมโครซอฟท์เป็นตัวรับรู้ท่าทางเพื่อรับอินพุตจากผู้ใช้ในการโต้ตอบกับโปรแกรมสร้างมโนภาพเชิงโต้ตอบ เนื่องจากมีคุณลักษณะในการรับรู้การเคลื่อนไหว ท่าทางของผู้ใช้ และสามารถเชื่อมต่อเพื่อใช้งานร่วมกับโปรแกรม processing ได้ โดยราคาตัวละ 3,500 บาท มีส่วนประกอบหลักดังแสดงในรูปที่ 2.16

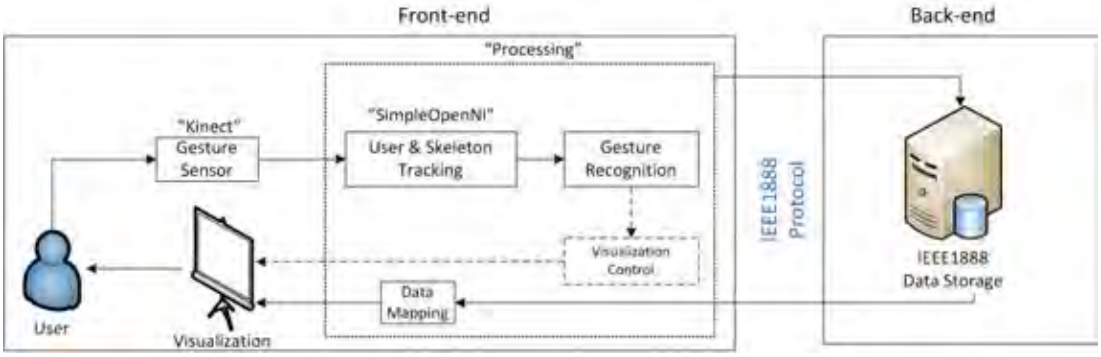
รูปที่ 2.16: ส่วนประกอบของตัวรับรู้ท่าทาง kinect

- **กล้องอาร์จีบี (RGB camera)** : เป็นกล้องเพื่อจับภาพสี มีความละเอียดในการวัดอยู่ที่ 640x480 จุดภาพ ที่อัตราเร็ว 30 เฟรมต่อวินาที
- **ตัวจ่ายแสงอินฟราเรด (IR emitter) และตัวรับรู้อินฟราเรด (IR sensor)** : ทำงานร่วมกันเพื่อทำหน้าที่เป็นตัวรับรู้ความลึก ระยะใช้งานอยู่ที่ 1.2 - 3.5 เมตร และความละเอียดในการวัดอยู่ที่ 640x480 จุดภาพ ที่อัตราเร็ว 30 เฟรมต่อวินาที
- **ไมโครโฟน (microphone)** : ส่งข้อมูลเสียงได้ด้วยความละเอียด 16 kHz
- **มอเตอร์เพื่อปรับการเอียง (tilt motor)** : ปรับความเอียงได้ในแนวตั้ง 27 องศา

บทที่ 3

สถาปัตยกรรมระบบและการทดสอบเบื้องต้น

3.1 สถาปัตยกรรมของระบบ



รูปที่ 3.1: สถาปัตยกรรมของระบบเพื่อพัฒนาโปรแกรมสร้างมโนภาพเชิงโต้ตอบ

สถาปัตยกรรมของระบบเพื่อใช้ในการพัฒนาโปรแกรมแบ่งออกเป็น 2 ส่วนหลัก ๆ คือ front-end และ back-end

- **front-end** ทำหน้าที่เป็นส่วนต่อประสานผู้ใช้ ประกอบไปด้วย
  1. **ตัวรับรู้ท่าทาง kinect** : ใช้รับอินพุตจากผู้ใช้งาน
  2. **คลัง (โปรแกรม) SimpleOpenNI** : ใช้ตรวจจับ และติดตามโครงกระดูกผู้ใช้งาน เพื่อใช้การแสดงท่าทางโต้ตอบกับโปรแกรม
  3. **โปรแกรม processing** : มีหน้าที่เรียกใช้คลัง (โปรแกรม) SimpleOpenNI, ร้องของข้อมูลจากหน่วยเก็บข้อมูล, ประมวลผลข้อมูล และทำการนำเสนอข้อมูล
  4. **หน้าจอแสดงผล** : ใช้ในการแสดงผลข้อมูล ซึ่งในงานวิจัยนี้ใช้โทรทัศน์จอแอลอีดีเป็นส่วนหน้าจอแสดงผล
- **back-end** ทำหน้าที่เป็นส่วนเก็บข้อมูลซึ่งในงานวิจัยนี้คือหน่วยเก็บข้อมูลของโครงการ CU-BEMS

เมื่อผู้ใช้งานเข้ามาในบริเวณที่ตัวรับรู้ท่าทาง kinect สามารถตรวจจับได้ kinect จะส่งอินพุตจากผู้ใช้ไปยัง SimpleOpenNI ซึ่งเป็นคลัง (โปรแกรม) ในโปรแกรม processing เพื่อตรวจจับผู้ใช้ซึ่งสามารถตรวจจับได้พร้อมกันสูงสุด 6 คน โดยแต่ละคนจะมีรหัสผู้ใช้ (userId) ของตนเอง และถูกลบออกเมื่อผู้ใช้งานไม่อยู่ในพื้นที่ตรวจจับ หลังจากตรวจจับผู้ใช้สำเร็จ SimpleOpenNI จะทำการติดตามโครงกระดูกของผู้ใช้งานแต่ละคนโดยอาศัยรหัสผู้ใช้ซึ่งจะทำให้โปรแกรม processing สามารถเรียกใช้ข้อมูลข้อต่อของโครงกระดูกเพื่อให้ผู้ใช้งานสามารถใช้ท่าทางในการควบคุมโปรแกรมสร้างมโนภาพได้ งานวิจัยนี้กำหนดให้ผู้ใช้ซึ่งมีรหัสผู้ใช้เป็นรหัสแรกเท่านั้นที่สามารถโต้ตอบกับโปรแกรมได้ สำหรับ

การติดต่อระหว่างส่วน front-end และ back-end ใช้โพรโทคอล FETCH จากโปรแกรม processing ใน front-end เพื่อร้องขอข้อมูลมายังหน่วยเก็บข้อมูลซึ่งอยู่ใน back-end และใช้โพรโทคอล WRITE จากโปรแกรม processing ใน front-end เพื่อเขียนค่าผู้เข้าใช้งานโปรแกรม และระยะเวลาในการใช้งานมายังหน่วยเก็บข้อมูลในส่วน back-end

3.2 การทดสอบเบื้องต้น

3.2.1 การทดสอบการทำงานของโปรแกรม processing

สำหรับคุณลักษณะของโปรแกรมในการบูรณาการณ์เข้ากับอุปกรณ์หรือระบบอื่น ๆ ได้ง่ายนั้น เช่น การเขียนโปรแกรมสำหรับการแสดงผลบนคอมพิวเตอร์ทั่วไป, บนเว็บ และบนอุปกรณ์ที่ใช้ระบบปฏิบัติการแอนดรอยด์ เป็นต้น ทำได้โดยการเขียนโปรแกรมแพียงครั้งเดียวจากนั้นเลือกรายการแบบวิธีที่บริเวณซ้ายมือด้านบนของส่วนต่อประสานหลักที่ต้องการดำเนินการโปรแกรม ในที่นี้ขอแสดงการใช้งานแบบวิธีต่าง ๆ อย่างง่ายโดยการเขียนโปรแกรมตามรูปที่ 3.2 ซึ่งกำหนดพื้นที่ทำงานขนาด 640x480 จุดภาพ มีพื้นหลังเป็นสีดำ จากนั้นวาดรูปสี่เหลี่ยม และวงกลมลงบนพื้นที่ ทดสอบการดำเนินงานโดยใช้แบบวิธีการต่าง ๆ ดังต่อไปนี้

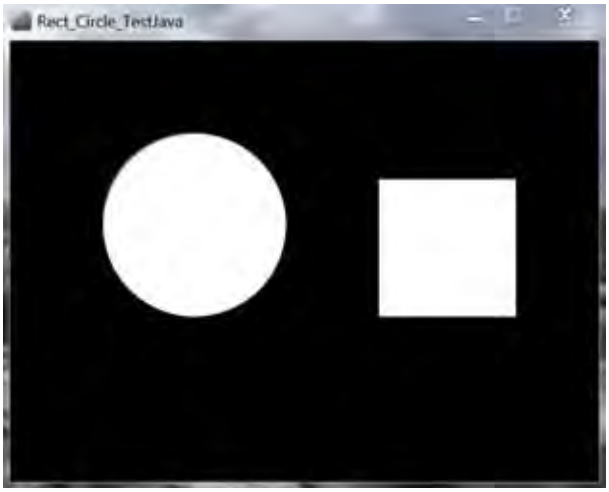
```
void setup(){
size(640,480);        ──────► กำหนดขนาดพื้นที่ทำงาน
background(0);        ──────► กำหนดสีของพื้นหลัง
}

void draw(){
  smooth();
  rect(400,150,150,150); ──────► วาดรูปสี่เหลี่ยม

  ellipse(200, 200,200,200); ──────► วาดรูปวงกลม
}
```

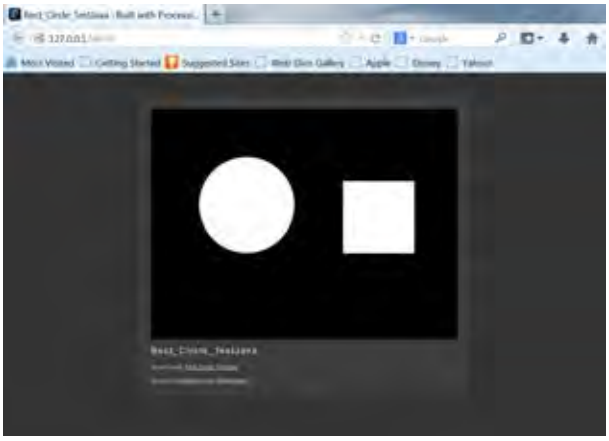
รูปที่ 3.2: โปรแกรมเพื่อทดสอบการดำเนินงาน

แบบวิธีจาวา (Java mode) เป็นแบบวิธีที่เป็นค่าปริยายของโปรแกรม ใช้สำหรับพัฒนาโปรแกรมเพื่อดำเนินการบนเครื่องคอมพิวเตอร์ทั่วไปในระบบปฏิบัติการต่าง ๆ รูปที่ 3.3 แสดงผลการดำเนินการเมื่อเขียนโปรแกรมเพื่อวาดรูปสี่เหลี่ยม และวงกลมบนหน้าจอแสดงผล โดยเลือกใช้แบบวิธีจาวา



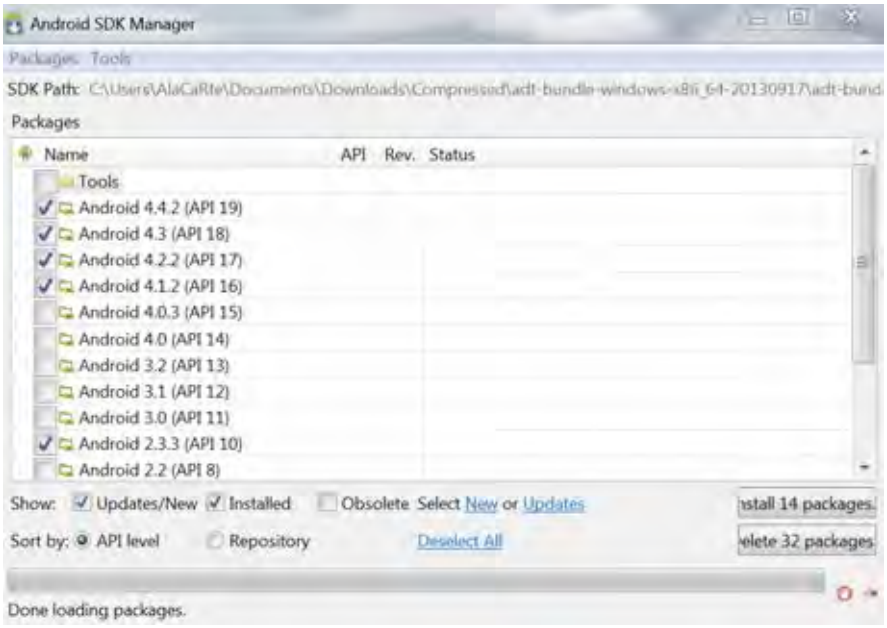
รูปที่ 3.3: การดำเนินการเมื่อเลือกใช้แบบวิธีจาวา

แบบวิธีจาวาสคริปต์ (JavaScript mode) เป็นแบบวิธีสำหรับการพัฒนาโปรแกรมที่จะนำไปใช้ดำเนินการบนเว็บ รูปที่ 3.4 แสดงผลการดำเนินการเมื่อเขียนโปรแกรมเพื่อวาดรูปสี่เหลี่ยม และวงกลมบนหน้าจอแสดงผล โดยเลือกใช้แบบวิธีจาวาสคริปต์



รูปที่ 3.4: การดำเนินการเมื่อเลือกใช้แบบวิธีจาวาสคริปต์

แบบวิธีแอนดรอยด์ (Android mode) เป็นแบบวิธีสำหรับการพัฒนาโปรแกรมที่จะนำไปใช้ดำเนินการบนอุปกรณ์ที่ใช้ระบบปฏิบัติการแอนดรอยด์ ก่อนเลือกใช้แบบวิธีนี้ ขั้นแรกจำเป็นต้องติดตั้งชุดพัฒนาซอฟต์แวร์แอนดรอยด์ (android SDK) ซึ่งสามารถดาวน์โหลดได้จากเว็บไซต์ http:/developer.android.com/sdk หลังจากติดตั้งชุดพัฒนาเรียบร้อยแล้วต้องทำการติดตั้งโปรแกรมสำเร็จ (package) ซึ่งอยู่ในตัวจัดการชุดซอฟต์แวร์แอนดรอยด์ตามรุ่นที่ต้องการพัฒนาโปรแกรม นอกจากนั้นในขั้นตอนนี้จำเป็นต้องติดตั้งเอพีไอระดับ 10 ร่วมด้วยทุกครั้งดังรูปที่ 3.5



รูปที่ 3.5: ตัวจัดการชุดพัฒนาซอฟต์แวร์แอนดรอยด์

ในส่วนของอุปกรณ์แอนดรอยด์ ผู้พัฒนาโปรแกรมจำเป็นต้องตั้งค่าอุปกรณ์ให้เป็นแบบวิธีการแก้จุดบกพร่อง (debugging mode) โดยในเข้าไปที่การตั้งค่า (setting) จากนั้นเลือกที่แบบวิธีผู้พัฒนา (developer mode) และเลือกแบบวิธีการแก้จุดบกพร่อง เมื่อทำครบทุกขั้นตอนแล้วนำสาย USB ของอุปกรณ์เสียบเข้ากับคอมพิวเตอร์และสั่งดำเนินการบนโปรแกรม processing ซึ่งผลจากการดำเนินการเมื่อเขียนโปรแกรเพื่อมวาดรูปสี่เหลี่ยม และวงกลมบนหน้าจอแสดงผลดังรูปที่ 3.6

รูปที่ 3.6: การดำเนินการเมื่อเลือกใช้แบบวิธีแอนดรอยด์

สำหรับวิทยานิพนธ์ผู้วิจัยนี้เลือกใช้แบบวิธีจาวาในการดำเนินการบนเครื่องคอมพิวเตอร์เนื่องจาก จำเป็นต้องเชื่อมต่อโปรแกรม processing เข้ากับตัวรับรู้ท่าทางซึ่งต้องอาศัยโปรแกรมขับของตัวรับรู้ที่ติดตั้งในคอมพิวเตอร์เครื่องเดียวกัน นอกจากนั้นยังต้องมีการประมวลผลข้อมูล รวมถึงการร้องขอ และเขียนข้อมูลไปยังหน่วยเก็บข้อมูลซึ่งไม่เหมาะกับการใช้แบบวิธีอื่น ๆ

3.2.2 การทดสอบการทำงานร่วมกันระหว่างตัวรับรู้ท่าทาง kinect และ โปรแกรม processing

ผู้วิจัยได้ทดสอบด้วยการเขียนโปรแกรม processing ดังรูปที่ 3.7 ภายในโปรแกรมเริ่มจากการให้โปรแกรม processing นำเข้าคลัง (โปรแกรม) SimpleOpenNI ด้วยคำสั่ง *import SimpleOpenNI.*;* จากนั้นใช้คำสั่ง *SimpleOpenNI kinect;* ประกาศอ็อบเจกต์ของ SimpleOpenNI ที่มีชื่อว่า kinect ซึ่งจะใช้ในการเข้าถึงข้อมูลของตัวรับรู้ kinect เช่น สามารถเรียกฟังก์ชันภาพความลึก (depth image), ภาพสี (color image) รวมถึงข้อมูลโครงกระดูกของผู้ใช้งาน ในส่วนการตั้งค่า (setup) กำหนดให้พื้นที่ดำเนินการของโปรแกรมมีความกว้าง 1280 จุดภาพ และความสูง 480 จุดภาพ ด้วยคำสั่ง *size(640*2,480);* และทำการสร้างอ็อบเจกต์ kinect ที่ได้ประกาศไว้ในข้างต้นโดยใช้คำสั่ง *kinect = new SimpleOpenNI(this);* หลังจากนั้นทำการเรียกเมท็อด *enableRGB*, *enableDepth* และ *enableUser* เพื่อแจ้งคลัง (โปรแกรม) ล่วงหน้าว่าเราต้องการเข้าถึงข้อมูลเหล่านี้เป็นอันจบขั้นตอนการตั้งค่าโปรแกรมเข้าสู่ลูปการทำงานของโปรแกรมโดยเริ่มจากการเรียกฟังก์ชัน update จากอ็อบเจกต์ kinect ด้วยคำสั่ง *kinect.update();* เพื่อให้โปรแกรมดึงค่าใหม่จากตัวรับรู้ kinect จากนั้นในส่วนของคำสั่ง *image(kinect.userImage(),0,0)* และ *image(kinect.rgbImage(),640,0)* เป็นการเรียกหาค่าล่าสุดของภาพผู้ใช้ และภาพสีตามลำดับ จากนั้นส่งภาพมายังฟังก์ชันการแสดงผลภาพของโปรแกรม processing ให้แสดงภาพผู้ใช้ในตำแหน่ง (0,0) และภาพสีในตำแหน่ง (640,0)

```
import SimpleOpenNI.*;

SimpleOpenNI kinect;

void setup() {
  size(640*2, 480);
  kinect = new SimpleOpenNI(this);
  kinect.enableRGB();
  kinect.enableDepth();
  kinect.enableUser();
}

void draw() {
  // delay(1000);
  kinect.update();
  image(kinect.userImage(), 0, 0);
  image(kinect.rgbImage(), 640, 0);
}
```

รูปที่ 3.7: โปรแกรมทดสอบการตรวจจับผู้ใช้

ผลจากการทดสอบด้วยโปรแกรมที่กล่าวในข้างต้นเป็นดังรูปที่ 3.8 เป็นทดลองดำเนินการโปรแกรมภายในห้องปฏิบัติการวิจัยระบบโทรคมนาคม ชั้น 13 อาคารเจริญวิศวกรรม ซึ่งทางด้านซ้ายของโปรแกรมแสดงภาพผู้ใช้ และด้านขวาแสดงภาพสีจากตัวรับรู้ท่าทาง kinect

รูปที่ 3.8: ผลการทดสอบการตรวจจับผู้ใช้

เมื่อทดสอบการตรวจจับผู้ใช้สำเร็จผู้วิจัยได้เขียนโปรแกรมเพื่อใช้ติดตามโครงกระดูกดังรูปที่ 3.9 และรูปที่ 3.10 ซึ่งเป็นการนำโปรแกรมเดิมที่ได้ทดสอบในข้างต้นมาเขียนเพิ่มเติม ขอเริ่มอธิบายจากส่วนท้ายสุดของโปรแกรม คือ ส่วนเฝ้ารอเหตุการณ์ (event listener) ในคำสั่ง *onNewUser(SimpleOpenNI test, int userId)* โดยเมื่อโปรแกรม SimpleOpenNI สามารถตรวจจับผู้ใช้จะมีการสร้างรหัสให้ผู้ใช้ (userId) ซึ่งเมื่อมีการสร้างรหัสผู้ใช้ ตัวเฝ้ารอเหตุการณ์จะทำงานในส่วนถัดไปของโปรแกรมคือ *test.startTrackingSkeleton(userId)* เป็นการเริ่มทำการติดตามโครงกระดูกของรหัสผู้ใช้ที่ได้รับ

ขอกลับมาในส่วนลูปการทำงานหลักของโปรแกรม จากการทดลองข้างต้นในการแสดงภาพผู้ใช้งานและภาพสีจากตัวรับรู้ท่าทาง kinect ผู้วิจัยเขียนโปรแกรมเพิ่มเติมโดยสร้างแถวลำดับ (array) ซึ่งชื่อว่า userList เพื่อเก็บรหัสผู้ใช้ที่ได้จากการเรียกใช้เมท็อด *getUsers()* ของอ็อบเจกต์ kinect จากนั้นเป็นการเช็คว่ามีผู้ใช้หรือไม่ด้วยคำสั่ง *if (userList.length>0)* ซึ่งถ้ามีผู้ใช้ให้ดึงรหัสผู้ใช้ในตำแหน่งแรกของแถวลำดับมาเป็นตั้งเป็นรหัสผู้ใช้ ตามคำสั่ง *int userId = userList[0]* ในตอนนี้รหัสผู้ใช้งานจะถูกตั้งให้เป็นค่าที่อยู่ในตำแหน่งแรกสุดของแถวลำดับ จากนั้นเป็นการเช็คว่ารหัสผู้ใช้งานนี้มีการติดตามโครงกระดูกสำเร็จแล้วหรือไม่ ด้วยคำสั่ง *if (kinect.isTrackingSkeleton(userId))* ซึ่งถ้าในส่วนตัวเฝ้ารอเหตุการณ์มีการติดตามโครงกระดูกสำเร็จ ในตอนนี้โปรแกรมจะพร้อมเรียกใช้ข้อมูลโครงกระดูกของผู้ใช้ได้แล้ว จากนั้นเป็นขั้นตอนการวาดโครงกระดูกลงบนภาพผู้ใช้โดยเริ่มจากการกำหนดสี และขนาดเส้นของโครงกระดูกด้วยคำสั่ง *stroke* และ *strokeWeight* ตามลำดับ ขั้นตอนสุดท้ายให้ทำการวาดโครงกระดูกของรหัสผู้ใช้แรกโดยใช้ฟังก์ชัน *drawSkeleton(userId)* ซึ่งใช้เมท็อด *drawLimb* เพื่อวาดเส้นเชื่อมต่อระหว่างข้อต่อต่าง ๆ ของโครงกระดูก

```
import SimpleOpenNI.*;

SimpleOpenNI kinect ;

void setup(){
  size(640*2, 480);
  kinect = new SimpleOpenNI(this);
  kinect.enableRGB();
  kinect.enableDepth();
  kinect.enableUser();
}

void draw(){
  kinect.update();
  image(kinect.userImage(), 0, 0);
  image(kinect.rgbImage(), 640, 0);

  int[] userList = kinect.getUsers();
  if (userList.length>0) {
    int userId = userList[0];
    if (kinect.isTrackingSkeleton(userId)) {
      stroke(255, 0, 0);
      strokeWeight(4);
      drawSkeleton(userId);
    }
  }
}
// draw skeleton
void drawSkeleton(int userId) {
```

รูปที่ 3.9: โปรแกรมทดสอบการติดตามโครงกระดูก

```
// draw skeleton
void drawSkeleton(int userId) {

  kinect.drawLimb(userId, SimpleOpenNI.SKEL_HEAD, SimpleOpenNI.SKEL_NECK);

  kinect.drawLimb(userId, SimpleOpenNI.SKEL_NECK, SimpleOpenNI.SKEL_LEFT_SHOULDER);
  kinect.drawLimb(userId, SimpleOpenNI.SKEL_LEFT_SHOULDER, SimpleOpenNI.SKEL_LEFT_ELBOW);
  kinect.drawLimb(userId, SimpleOpenNI.SKEL_LEFT_ELBOW, SimpleOpenNI.SKEL_LEFT_HAND);

  kinect.drawLimb(userId, SimpleOpenNI.SKEL_NECK, SimpleOpenNI.SKEL_RIGHT_SHOULDER);
  kinect.drawLimb(userId, SimpleOpenNI.SKEL_RIGHT_SHOULDER, SimpleOpenNI.SKEL_RIGHT_ELBOW);
  kinect.drawLimb(userId, SimpleOpenNI.SKEL_RIGHT_ELBOW, SimpleOpenNI.SKEL_RIGHT_HAND);

  kinect.drawLimb(userId, SimpleOpenNI.SKEL_LEFT_SHOULDER, SimpleOpenNI.SKEL_TORSO);
  kinect.drawLimb(userId, SimpleOpenNI.SKEL_RIGHT_SHOULDER, SimpleOpenNI.SKEL_TORSO);

  kinect.drawLimb(userId, SimpleOpenNI.SKEL_TORSO, SimpleOpenNI.SKEL_LEFT_HIP);
  kinect.drawLimb(userId, SimpleOpenNI.SKEL_LEFT_HIP, SimpleOpenNI.SKEL_LEFT_KNEE);
  kinect.drawLimb(userId, SimpleOpenNI.SKEL_LEFT_KNEE, SimpleOpenNI.SKEL_LEFT_FOOT);

  kinect.drawLimb(userId, SimpleOpenNI.SKEL_TORSO, SimpleOpenNI.SKEL_RIGHT_HIP);
  kinect.drawLimb(userId, SimpleOpenNI.SKEL_RIGHT_HIP, SimpleOpenNI.SKEL_RIGHT_KNEE);
  kinect.drawLimb(userId, SimpleOpenNI.SKEL_RIGHT_KNEE, SimpleOpenNI.SKEL_RIGHT_FOOT);
}

void onNewUser(SimpleOpenNI test, int userId)
{
  println("onNewUser - userId: " + userId);
  println("tracking skeleton");
  test.startTrackingSkeleton(userId);
}
```

รูปที่ 3.10: โปรแกรมทดสอบการติดตามโครงกระดูก (ต่อ)

ผลจากการทดสอบที่เขียนโปรแกรมติดตามโครงการดูกเป็นดังรูปที่ 3.11 ทำการทดลองดำแนินการโปรแกรมในห้องปฏิบัติการวิจัยระบบโทรคมนาคม ชั้น 13 อาคารเจริญวิศวกรรม ซึ่งทางด้านซ้ายของโปรแกรมแสดงภาพผู้ใช้พร้อมทั้งวาดโครงกระดูกลงบนภาพผู้ใช้ และด้านขวาแสดงภาพสีจากตัวรับรู้ท่าทาง kinect

รูปที่ 3.11: ผลการทดสอบการตรวจจับผู้ใช้ และการติดตามโครงกระดูก

หลังจากโปรแกรมสามารถตรวจจับผู้ใช้ และติดตามโครงกระดูกสำเร็จแล้วเราจะสามารถเรียกใช้ข้อมูลข้อต่อของโครงกระดูกจากคลัง (โปรแกรม) SimpleOpenNI ได้แล้ว ซึ่งผู้วิจัยจะนำค่าที่ได้ไปใช้เป็นอินพุตของผู้ใช้เพื่อตอบโต้กับโปรแกรมสร้างมโนภาพที่พัฒนาขึ้น และจากการทดสอบทั้งสองขั้นตอนในข้างต้นตัวรับรู้ท่าทาง kinect สามารถทำงานได้ดีเมื่อผู้ใช้งานอยู่ห่างจากตัวรับรู้ท่าทาง kinect ประมาณ 1.5 เมตร

3.2.3 ทดสอบการเชื่อมต่อระหว่างโปรแกรม processing กับหน่วยเก็บข้อมูลบนพื้นฐาน IEEE1888 โดยใช้โปรแกรม PHP

ขั้นตอนนี้ใช้สำหรับการทดสอบการเชื่อมต่อของส่วน front-end และ back-end โดยเขียนโปรแกรม PHP เพื่อรองรับการร้องขอข้อมูลจากโปรแกรม processing ไปยังหน่วยเก็บข้อมูลด้วยโพรโตคอล FETCH โดยโปรแกรมที่ใช้ในการร้องขอข้อมูลจากโปรแกรม processing แสดงดังรูปที่ 3.12 - 3.14 เริ่มจากโปรแกรม processing ร้องขอไปยังโปรแกรม PHP เพื่อทำการ FETCH โดยส่งข้อความร้องขอเป็นสายอักขระที่ระบุตำแน่ง, ชนิดของตัวรับรู้, และช่วงเวลาที่สนใจ โดยในการทดสอบต้องการข้อมูลของอุณหภูมิซึ่งติดตั้งอยู่ภายในห้องวิจัยระบบสื่อสารที่อยู่ในชั้น 13 ของอาคาร 4 ในช่วงวันที่ 14 มีนาคม 2014 ตั้งแต่เวลา 7:00 ถึง 10:05 และนำค่าที่ได้รับมาเก็บไว้ยังแถวลำดับชนิดสายอักขระตามคำสั่ง

```
String[] php = loadStrings("http://161.200.90.5/atest1888/A1_Dymanic.php"+
"?Building=eng4&Floor=fl13&Zone=lab_commu&Sensor=temperature"+
"&sYear=2014&sMonth=03&sDate=14&sHour=07&sMin=00"+
"&eYear=2014&eMonth=03&eDate=14&eHour=10&eMin=05");,


///>>>>>>>>>>>>> For SPLIT PHP TO YEAR&DATE&TEMPERATURE >>>>>>>>>>>>>>
///>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
ArrayList<String> Timestamp = new ArrayList<String>();
ArrayList<Float> Temperature = new ArrayList<Float>();

int i = 0;
boolean stopCondition = false;

while (stopCondition == false) {

  String[] sep = split((php[i]), 't');

  stopCondition = sep[0].equals("END");

  if (stopCondition == false) {
    Timestamp.add(sep[0]); // Timestamp contains YEAR MONTH DATE & H M S
    Temperature.add(float(sep[1])); //  contains TEMPERATURE
   // println(php[i]);
    //println(sep);
    i = i+1;
  }
}
```

รูปที่ 3.12: โปรแกรมทดสอบการร้องขอข้อมูลไปยังหน่วยเก็บข้อมูล

```
///>>>>>>>>>>>>>>>>>>>>>>>> For SPLIT  TO YMD & HMS >>>>>>>>>>>>>>
///>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
ArrayList<String> Timestamp0 = new ArrayList<String>();
ArrayList<String> Timestamp1 = new ArrayList<String>();
for(int j = 0; j<(Timestamp.size());j++){
 String[] stamp = split(Timestamp.get(j),'T');
 Timestamp0.add(stamp[0]); // AAA contains YMD
 Timestamp1.add(stamp[1]); // BBB contains HMS
}

//>>>>>>>>>>>>>>>>>>>> For SPLIT Timestamp0 TO Y M D >>>>>>>>>>>>>>>
//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

ArrayList<Integer> YEAR = new ArrayList<Integer>();
ArrayList<Integer> MONTH = new ArrayList<Integer>();
ArrayList<Integer> DATE = new ArrayList<Integer>();
for(int k = 0; k<(Timestamp0.size()); k++){
  int[] ymd = int(split((Timestamp0.get(k)), '-'));
  YEAR.add(ymd[0]);
  MONTH.add(ymd[1]);
  DATE.add(ymd[2]);
}

//println(DATE.get(200));
//>>>>>>>>>>>>>>>>>>>> For SPLIT Timestamp1 TO H Min S >>>>>>>>>>>>>>
//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
```

รูปที่ 3.13: โปรแกรมทดสอบการร้องขอข้อมูลไปยังหน่วยเก็บข้อมูล(ต่อ)

```
//>>>>>>>>>>>>>>>>>>>> For SPLIT Timestamp1 TO H Min S >>>>>>>>>>>>>>
//>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
ArrayList<Integer> HOUR = new ArrayList<Integer>();
ArrayList<Integer> MIN = new ArrayList<Integer>();
ArrayList<Integer> SEC = new ArrayList<Integer>();
for(int L = 0; L<(Timestamp1.size());L++){
  int[] hms = int(split((Timestamp1.get(L)), ':'));
  HOUR.add(hms[0]);
  MIN.add(hms[1]);
  SEC.add(hms[2]);
}
println("Data Loaded: "+php.length+" entries.");
println("Data Useed: "+Temperature.size()+" entries.");
println(YEAR.get(15));
println(MONTH.get(15));
println(DATE.get(15));
println(HOUR.get(15));
println(MIN.get(15));
println(SEC.get(15));
println(Temperature.get(15));
println(php[15]);
```

รูปที่ 3.14: โปรแกรมทดสอบการร้องขอข้อมูลไปยังหน่วยเก็บข้อมูล(ต่อ)

จากนั้นสร้างส่วนต่อประสานกับผู้ใช้อย่างง่ายซึ่งสามารถเลือกข้อมูล, บริเวณ และช่วงเวลาที่สนใจเพื่อทดสอบการเชื่อมต่อ และทดสอบการสร้างมโนภาพเบื้องต้นด้วยการแสดงผลข้อมูลเป็นกราฟเส้น จากรูปที่ 3.15 แสดงส่วนต่อประสานผู้ใช้ที่สร้างโดยโปรแกรม processing และร้องขอข้อมูลอุณหภูมิบริเวณห้องปฏิบัติการวิจัยระบบโทรคมนาคม ในวันที่ 14 มีนาคม 2014 ตั้งแต่เวลา 7:00 ถึง 10:00

รูปที่ 3.15: ผลการทดสอบการเชื่อมต่อระหว่างโปรแกรม processing กับหน่วยเก็บข้อมูลบนพื้นฐาน IEEE1888 โดยใช้โปรแกรม PHP

3.2.4 การทดสอบการเชื่อมต่อระหว่างโปรแกรม processing กับหน่วยเก็บข้อมูลบนพื้นฐาน IEEE1888 โดยตรง

ในหัวข้อก่อนหน้านี้ใช้โปรแกรม PHP ในการทดสอบร้องขอข้อมูลจากหน่วยเก็บข้อมูลเพื่อนำมาแสดงผลอย่างง่าย แต่อย่างไรก็ตามการสื่อสารที่ใช้ยังไม่เป็นการสื่อสารโดยตรงตามมาตรฐาน IEEE1888 ในการทดสอบนี้ผู้วิจัยจึงเขียนโปรแกรมเพื่อใช้สำหรับสื่อสารโดยตรงระหว่างโปรแกรม processing และหน่วยเก็บข้อมูลของโครงการ CU-BEMS ให้เป็นไปตามมาตรฐาน IEEE1888 ซึ่งในหัวข้อนี้นี้เป็นการทดสอบร้องขอข้อมูลตามรูปแบบของโพรโทคอล FETCH และส่งข้อมูลตามรูปแบบของโพรโทคอล WRITE โดยมีการตรวจสอบรูปแบบการสื่อสารด้วยโปรแกรม wireshark

3.2.4.1 การทดสอบการร้องขอข้อมูลโดยใช้โพรโทคอล FETCH

ทดสอบร้องขอข้อมูลไปยัง URI ซึ่งมีค่าข้อมูลการใช้พลังงานรวมของทั้งภาควิชาวิศวกรรมไฟฟ้า id = "bems.ee.eng.chula.ac.th/energy_consumption/department" มีรูปแบบของข้อมูลที่ใช้ส่งเพื่อทำการร้องขอเมื่อใช้โปรแกรม wireshark ตรวจจับเป็นไปตามรูปที่ 3.16 และหน่วยเก็บข้อมูลจะตอบกลับมาเป็นรูปแบบดังแสดงในรูปที่ 3.17

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
 <soapenv:Body>
  <ns2:queryRQ
    xmlns:ns2="http://soap.fiap.org/">
   <transport
     xmlns="http://gutp.jp/fiap/2009/11/">
    <header>
     <query
       id="9eed9de4-1c48-4b08-a41d-dac067fc1c0d"
       type="storage">
      <key
        id="bems.ee.eng.chula.ac.th/energy_consumption/department"
        attrName="time"
        select="maximum"/>
      </query>
     </header>
    </transport>
   </ns2:queryRQ>
  </soapenv:Body>
 </soapenv:Envelope>
```

รูปที่ 3.16: การร้องขอข้อมูลจาก processing ไปยังหน่วยเก็บข้อมูลโดยโพรโทคอล FETCH

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
 <soapenv:Body>
  <ns2:queryRS
    xmlns:ns2="http://soap.fiap.org/">
   <transport
     xmlns="http://gutp.jp/fiap/2009/11/">
    <header>
     <OK/>
     <query
       id="9eed9de4-1c48-4b08-a41d-dac067fc1c0d"
       type="storage">
      <key
        id="bems.ee.eng.chula.ac.th/energy_consumption/department"
        attrName="time"
        select="maximum"/>
       </query>
      </header>
    <body>
     <point
       id="bems.ee.eng.chula.ac.th/energy_consumption/department">
      <value
        time="2014-10-20T23:05:00.000+07:00">
        20897.567143
        </value>
       </point>
      </body>
     </transport>
    </ns2:queryRS>
   </soapenv:Body>
  </soapenv:Envelope>
```


รูปที่ 3.17: การตอบกลับพร้อมทั้งข้อมูลจากหน่วยเก็บข้อมูลเมื่อใช้โพรโทคอล FETCH

3.2.4.2 การทดสอบการส่งข้อมูลโดยใช้โพรโทคอล WRITE

ทดสอบส่งข้อมูลไปที่ PointSet ID = "http://preecha.test.chula.ac.th/processing/kinect1" โดยข้อมูลที่ส่งไปนั้นประกอบด้วยจำนวนผู้ใช้งาน และระยะเวลาที่ผู้ใช้งานเข้าใช้งานระบบซึ่งมี point ID ดังนี้

- "http://preecha.test.chula.ac.th/processing/kinect1/userNumber"
- "http://preecha.test.chula.ac.th/processing/kinect1/userspentTime"

หลังจากนั้นใช้โปรแกรม wireshark ตรวจจับรูปแบบของข้อมูลที่โปรแกรม processing ส่งไปยังหน่วยเก็บข้อมูลพบว่าเป็นไปตามรูปที่ 3.18 และหน่วยเก็บข้อมูลจะตอบกลับมาเป็นรูปแบบดังแสดงในรูปที่ 3.19

```
<soapenv:Envelope
    xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns2:dataRQ
        xmlns:ns2="http://soap.fiap.org/">
      <transport
          xmlns="http://gutp.jp/fiap/2009/11/">
        <body>
          <pointSet
              id="http://preecha.test.chula.ac.th/processing/kinect1">
            <point
                id="http://preecha.test.chula.ac.th/processing/kinect1/userNumber">
              <value
                  time="2557-10-21T15:19:09.000+07:00">
                  28
                  </value>
              </point>
            <point
                id="http://preecha.test.chula.ac.th/processing/kinect1/userspentTime">
              <value
                  time="2557-10-21T15:19:09.000+07:00">
                  68
                  </value>
              </point>
            </pointSet>
          </body>
        </transport>
      </ns2:dataRQ>
    </soapenv:Body>
  </soapenv:Envelope>
```

รูปที่ 3.18: การส่งข้อมูลจาก processing ไปยังหน่วยเก็บข้อมูลเมื่อใช้โพรโทคอล WRITE

```
<soapenv:Envelope
    xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns2:dataRS
        xmlns:ns2="http://soap.fiap.org/">
      <transport
          xmlns="http://gutp.jp/fiap/2009/11/">
        <header>
            <OK/>
            </header>
          </transport>
        </ns2:dataRS>
      </soapenv:Body>
    </soapenv:Envelope>
```

รูปที่ 3.19: การตอบกลับจากหน่วยเก็บข้อมูลเมื่อใช้โพรโทคอล WRITE

3.3 การพัฒนา และวิธีการประเมินโปรแกรมต้นแบบ

ก่อนทำการพัฒนาโปรแกรมสร้างมโนภาพเพื่อการใช้งานจริงในโครงการ CU-BEMS ผู้วิจัยได้ทำการพัฒนาโปรแกรมต้นแบบโดยใช้เกมส์คอมพิวเตอร์ในการนำเสนอข้อมูลเพื่อดึงดูดความสนใจของผู้เล่น และทำการประเมินโปรแกรมโดยใช้แบบสอบถามเพื่อวัดความพึงพอใจในการใช้งาน

3.3.1 โปรแกรมต้นแบบที่ถูกพัฒนาขึ้น

รูปแบบของเกมส์ที่นำมาใช้ทำโปรแกรมต้นแบบนั้นใช้แนวคิดของเกมส์ 2048 ซึ่งเป็นที่นิยมอย่างมาก โปรแกรมต้นแบบได้มีการดัดแปลงรูปแบบของเกมส์ 2048 มาตรฐานซึ่งมีขนาด 4X4 ให้มีขนาดเป็น 3X3 เพื่อความพอเหมาะของระยะเวลาที่ใช้ในการเล่นเกมส์แต่ละครั้ง ข้อมูลการใช้พลังงาน ณ วันปัจจุบันถูกนำมาใช้เป็นค่าเริ่มต้นของเกมส์ซึ่งจะสุ่มจุดตั้งต้นในแต่ละครั้งของการเล่นเกมส์ ผู้เล่นจะต้องใช้มือขวาในการเลื่อนเพื่อรวมวงกลมที่มีค่าเท่ากันเพื่อให้ได้คะแนนเพิ่มมากขึ้น เกมส์จะจบลงเมื่อผู้เล่นไม่สามารถย้ายค่าใด ๆ เพิ่มเติมได้อีก เนื่องจากมีค่าเต็มทุกช่องแล้ว จากรูปที่ 3.20 แสดงภาพหน้าจอของโปรแกรมต้นแบบซึ่งถูกพัฒนาขึ้น ซึ่งมีค่าการใช้พลังงาน 2,202 กิโลวัตต์ต่อชั่วโมง ถ่ายทอดสู่ผู้เล่นเกมส์ซ้ำ ๆ ตลอดการเล่นเกมส์

รูปที่ 3.20: โปรแกรมต้นแบบ energy game

3.3.2 วิธีการประเมินโปรแกรมต้นแบบที่ถูกพัฒนาขึ้น

ในการประเมินโปรแกรมต้นแบบที่พัฒนาขึ้นนั้น ผู้วิจัยได้เชิญผู้เข้าร่วมทดลองใช้งานจำนวน 30 คน รายละเอียดของผู้เข้าร่วมทดลองใช้งานโปรแกรมต้นแบบแสดงในตารางที่ 3.1 ผู้เข้าร่วมทดลองส่วนใหญ่เป็นสมาชิกภายในห้องปฏิบัติการวิจัยเนื่องจากจุดติดตั้งสำหรับใช้งานจริงอยู่บริเวณโถงลิฟท์หน้าห้องปฏิบัติการวิจัย ในส่วนข้อมูลจากบุคคลทั่วไปภายนอกเพื่อให้ได้ข้อมูลสำหรับการปรับปรุงในเรื่องของความง่ายในการใช้งาน

ตารางที่ 3.1: รายละเอียดของผู้เข้าร่วมทดลองใช้งานโปรแกรมต้นแบบ

อายุ (ปี)	รายละเอียด	เปอร์เซนต์
22 - 26	นิสิตภายในห้องปฏิบัติการวิจัย	66.67 %
26 - 60	บุคคลทั่วไปจากภายนอก	33.33 %

ผู้เข้าร่วมทำการทดลองแต่ละคนถูกขอให้เล่นเกมส์คนละ 1 รอบ โดยโปรแกรมต้นแบบที่พัฒนาขึ้นติดตั้งอยู่บนคอมพิวเตอร์ซึ่งมีคุณลักษณะดังแสดงในตารางที่ 3.2 และแสดงผลบนหน่วยแสดงผลภายนอกขนาด 20 นิ้ว ซึ่งตั้งอยู่บริเวณโต๊ะภายในห้องปฏิบัติการวิจัย

ตารางที่ 3.2: คุณลักษณะของคอมพิวเตอร์โน๊ตบุ๊คที่ใช้ในการทดสอบโปรแกรมต้นแบบ

model	description
Sony Vaio VGN-FW46SJ	intel core2 duo processor T9600 2.80 GHz processor 4GB DDR2 SDRAM windows 8 professional 64 bit operating system

หลังจากทดลองใช้งานโปรแกรมต้นแบบ ผู้เข้าร่วมทำการทดลองถูกขอให้ตอบแบบสอบถามเพื่อใช้ในการประเมินผลตามตางรางที่ 3.3 โดยคำถามที่ใช้นำมาจากแบบสอบถามการประเมินผลโปรแกรมต่อประสานซึ่งเป็นที่นิยม การประเมินความพึงพอใจของผู้ใช้ได้นำ Likert scale [22] มาใช้ โดยในแต่ละคำถามคะแนนจะเริ่มจาก 1 เห็นด้วยอย่างยิ่ง ไปจนถึง 5 ไม่เห็นด้วยอย่างยิ่ง ตามลำดับ ผลการประเมินความพึงพอใจแสดงในตารางที่ 3.4 ในแง่ของค่าเฉลีย และส่วนเบี่ยงเบนมาตรฐานของกลุ่มตัวอย่างของคะแนนในแต่ละคำถาม ซึ่งผลจากการประเมินความพึงพอใจนั้นค่าเฉลี่ยของความพึงพอใจจากผู้ใช้งานเป็นที่ยอมรับได้

ตารางที่ 3.3: แบบสอบถามสำหรับประเมินความพึงพอใจโปรแกรมต้นแบบที่ถูกพัฒนาขึ้น

คำถาม	**ความเห็น**
1	มันมีประโยชน์
2	มันเป็นเรื่องง่ายที่จะใช้โปรแกรมต่อประสานนี้
3	ฉันสามารถใช้งานได้โดยไม่ต้องอ่านคำแนะนำ
4	ฉันสามารถใช้งานได้สำเร็จทุกครั้ง
5	มันเป็นเรื่องง่ายที่จะเรียนรู้การใช้งานโปรแกรมต่อประสานนี้
6	ข้อมูลที่มีให้ในระบบง่ายต่อการทำความเข้าใจ
7	ฉันชอบใช้โปรแกรมต่อประสานของระบบนี้
8	มันสนุกที่จะใช้โปรแกรมต่อประสานนี้
9	ฉันจะแนะนำโปรแกรมต่อประสานนี้ให้กับเพื่อน
10	โดยรวมแล้วฉันพอใจกับความง่ายในการใช้งานของระบบนี้

ตารางที่ 3.4: ผลคะแนนการแสดงความเห็นของคำถามในแต่ละข้อ

คำถาม	ค่าเฉลี่ยของกลุ่มตัวอย่าง	ส่วนเบี่ยงเบนมาตรฐานของกลุ่มตัวอย่าง
1	2.30	0.53
2	2.00	0.52
3	2.43	0.81
4	2.26	0.45
5	2.51	0.68
6	1.81	0.55
7	1.66	0.84
8	1.59	0.50
9	1.85	0.75
10	1.92	0.69

3.4 ข้อสรุป

ผลจากการทดสอบการเบื้องต้นได้ข้อสรุปว่าการทำงานร่วมกันระหว่างส่วนประกอบต่าง ๆ ภายในระบบเป็นไปตามหน้าที่ ในส่วนของผลการพัฒนาโปรแกรมต้นแบบเพื่อทดสอบการบูรณาการระบบและประเมินความพึงพอใจของผู้ใช้งานพบว่าค่าเฉลี่ยของความพึงพอใจจากผู้ใช้งานเป็นที่ยอมรับได้พร้อมที่จะใช้ข้อมูลที่ได้ศึกษาในข้างต้นมาใช้เพื่อพัฒนาโปรแกรมสำหรับใช้งานจริงในโครงการ CU-BEMS นอกจากการประเมินโปรแกรมโดยใช้แบบสอบถามแล้วในการใช้งานจริงจะมีการพัฒนาสำหรับการประเมินผลเชิงปริมาณเพิ่มเติมซึ่งจะกล่าวในบทถัดไป

บทที่ 4

การพัฒนาโปรแกรมสร้างมโนภาพเชิงโต้ตอบด้วยตัวรับรู้ท่าทางตามมาตรฐาน IEEE1888 เพื่อการใช้งานจริงในโครงการ CU-BEMS

ในบทที่ผ่านมากล่าวถึงภาพรวมของระบบ การทดสอบการทำงานของแต่ละส่วนเบื้องต้นและการพัฒนาต้นแบบเพื่อทดสอบการทำงานพื้นฐานของระบบ รวมถึงการประเมินโปรแกรมต้นแบบจากความพึงพอใจในการใช้งานของผู้เข้าร่วมทดสอบ เพื่อเตรียมสำหรับการพัฒนาโปรแกรมสร้างมโนภาพเชิงโต้ตอบด้วยตัวรับรู้ท่าทางตามมาตรฐาน IEEE1888 เพื่อการใช้งานจริงในโครงการ CU-BEMS ในบทนี้จะเป็นการกล่าวถึงการออกแบบส่วนแสดงผลหลัก และผลการติดตั้งเพื่อการใช้งานจริง

4.1 การออกแบบโปรแกรมสร้างมโนภาพ

การออกแบบโปรแกรมผู้วิจัยได้กำหนดคุณสมบัติหลักของโปรแกรมไว้ 3 ส่วนดังนี้

4.1.1 รูปแบบการนำเสนอ : ใช้การนำเสนอในเชิงทัศนศิลป์ (visual art)

การแสดงผลข้อมูลสำหรับโปรแกรมที่พัฒนาขึ้นผู้วิจัยเลือกนำศิลปะมาร่วมกับการนำเสนอโดยส่วนแสดงผลหลักของโปรแกรมประกอบได้ด้วย 5 แคนวาสซึ่งมีรูปแบบการแสดงผลที่สามารถแบ่งออกเป็นส่วนต่าง ๆ ดังนี้

แคนวาสแรกสุด (first canvas) : เป็นแคนวาสพักหน้าจอ ใช้การสุ่มวางภาพที่บรรจุในโฟลด์เดอร์ซึ่งถูกกำหนดไว้มาแสดงบนหน้าจอโดยการสุ่มวางภาพในจุดที่แตกต่างกัน องศาในการวางที่แตกต่างกัน และทำการเบลอภาพเก่าพร้อมทั้งเปลี่ยนให้เป็นสีเทาเพื่อเพิ่มความน่าสนใจของภาพล่าสุดที่ถูกวางลงบนหน้าจอดังแสดงในรูปที่ 4.1 สำหรับโฟลเดอร์เก็บข้อมูลภาพซึ่งใช้ในโครงการนี้ผู้วิจัยได้สร้างบัญชีผู้ใช้ของ dropbox ไว้เพื่อให้ผู้มีส่วนเกี่ยวข้องสามารถนำภาพบรรจุขึ้นในโฟลเดอร์ได้ผ่านโครงข่ายอินเทอร์เน็ต หน้าการทำงานนี้จะเปลี่ยนไปยังแคนวาสแสดงผลหลักของโปรแกรมเมื่อตัวรับรู้ท่าทางสามารถตรวจจับว่า มีผู้ใช้งานอยู่ในบริเวณหน้าจอแสดงผล

รูปที่ 4.1: หน้าจอแรกของโปรแกรมที่ถูกพัฒนาขึ้น

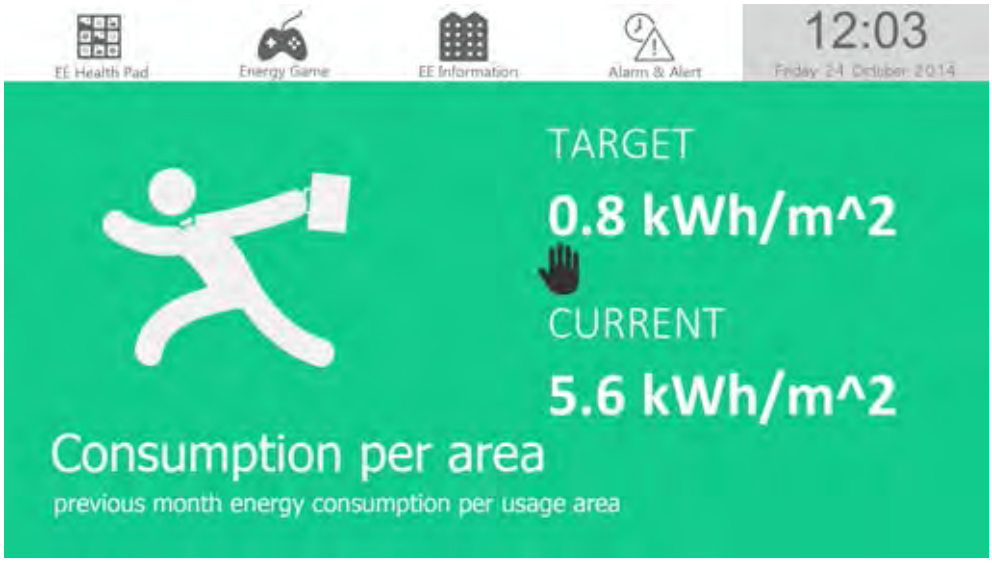
แคนวาสสรุปรวมสมรรถนะระบบของภาควิชาวิศวกรรมไฟฟ้า (EE health pad) : เป็นแคนวาสสรุปรวมสมรรถนะระบบของโครงการ CU-BEMS ซึ่งรวมดรรชนีชี้วัดระบบดังรูปที่ 4.2 เพื่อนำเสนอโดยแยกเป็น 9 หน้า ซึ่งแสดงข้อมูลดัชนีแต่ละตัว และได้เพิ่มการโต้ตอบระหว่างผู้ใช้งานและโปรแกรมแสดงผลโดยให้ผู้ใช้งานสามารถใช้การโบกมือไปทางซ้ายหรือขวาเพื่อเลื่อนดูข้อมูลดัชนีตัวก่อนหน้าหรือตัวถัดไปได้

รูปที่ 4.2: หน้าจอสรุปรวมดรรชนีชี้วัดระบบ CU-BEMS บนเว็บไวต์ของโครงการ

ส่วนแสดงผลหลักของของแคนวาสนี้ประกอบไปด้วย 9 หน้าจอดังแสดงในรูปที่ 4.3 - 4.11 ตามลำดับซึ่งมีรายละเอียดของค่าดรรชนีทั้ง 9 ค่าดังต่อไปนี้

1. ค่าความต้องการพลังงานสูงสุดที่ลดลงได้จากเกณฑ์สูงสุดที่ตั้งไว้
2. ค่าการใช้พลังงานต่อพื้นที่
3. ค่าโหลดแฟกเตอร์
4. ค่าไฟฟ้าที่ลดได้
5. ค่าพลังงานที่ลดได้
6. ค่าพลังงานที่ลดได้เทียบเป็นปริมาณการใช้ถ่านหินสำหรับผลิตไฟฟ้า
7. ค่าพลังงานที่ลดได้เทียบเป็นปริมาณการปล่อยก๊าซคาร์บอนไดออกไซด์ในการผลิตไฟฟ้า
8. ค่าพลังงานที่จ่ายจากแหล่งพลังงานหมุนเวียนในระบบ
9. ค่าพลังงานที่ลดได้เทียบเป็นจำนวนต้นไม้ที่ต้องปลูกเพื่อดูดซับก๊าซคาร์บอนไดออกไซด์

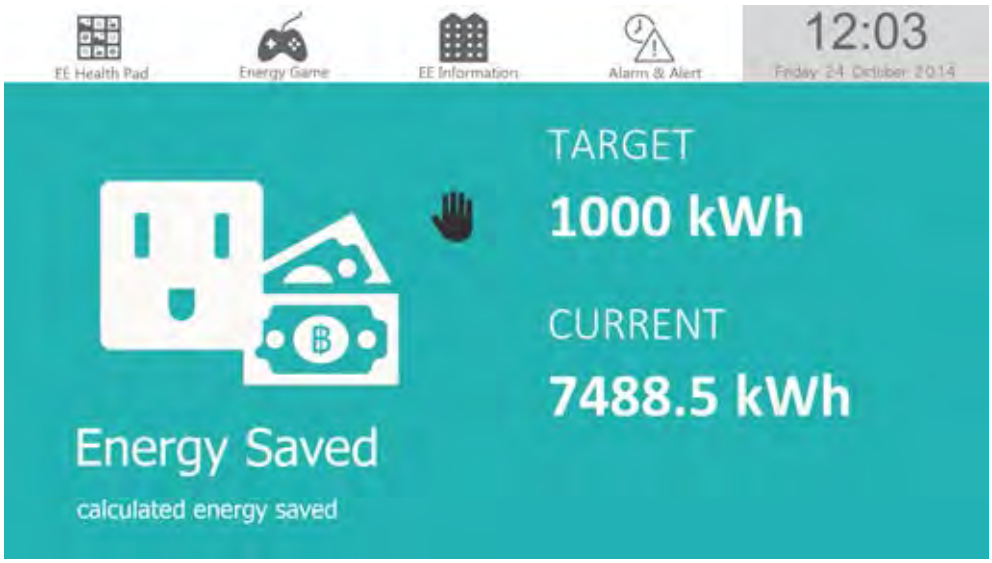
รูปที่ 4.3: หน้าจอแสดงค่าความต้องการพลังงานสูงสุดที่ลดลงได้จากเกณฑ์สูงสุดที่ตั้งไว้



รูปที่ 4.4: หน้าจอแสดงค่าการใช้พลังงานต่อพื้นที่

รูปที่ 4.5: หน้าจอแสดงค่าโหลดแฟกเตอร์

รูปที่ 4.6: หน้าจอแสดงค่าไฟฟ้าที่ลดได้



รูปที่ 4.7: หน้าจอแสดงค่าพลังงานไฟฟ้าที่ลดได้

รูปที่ 4.8: หน้าจอแสดงค่าพลังงานที่ลดได้เทียบเป็นปริมาณการใช้ถ่านหินสำหรับผลิตไฟฟ้า

รูปที่ 4.9: หน้าจอแสดงค่าพลังงานที่ลดได้เทียบเป็นปริมาณการปล่อยก๊าซคาร์บอนไดออกไซด์ในการผลิตไฟฟ้า

รูปที่ 4.10: หน้าจอแสดงค่าพลังงานที่จ่ายจากแหล่งพลังงานหมุนเวียนในระบบ

รูปที่ 4.11: หน้าจอแสดงค่าพลังงานที่ลดได้เทียบเป็นจำนวนต้นไม้

แคนวาสเกมส์แสดงการใช้พลังงานไฟฟ้า (energy game) : สำหรับแคนวาสนี้เป็นการนำข้อมูลการใช้พลังงานของแต่ละพื้นที่ภายในภาควิชามาใช้ในเกมส์ซึ่งพัฒนาเพิ่มเติมจากโปรแกรมต้นแบบ พื้นที่ซึ่งมีการใช้พลังงานน้อยที่สุดจะถูกนำมาเป็นค่าเริ่มต้นในการเริ่มเกมส์ดังรูปที่ 4.12

รูปที่ 4.12: หน้าจอเมื่อเริ่มเกมส์

พื้นที่ด้านขวาของหน้าจอแสดงสัญรูปของแต่ละพื้นที่โดยเรียงตั้งแต่ค่าเริ่มต้นของเกมส์จนถึงพื้นที่ซึ่งผู้เล่นสามารถทำคะแนนได้สูงสุด ณ ขณะนั้นพร้อมทั้งแสดงค่าการใช้พลังงานจริงของแต่ละพื้นที่ ดังแสดงในรูปที่ 4.13 ผู้เล่นต้องใช้มือขวาในการเลื่อนสัญรูปที่เหมือนเข้าด้วยกันเพื่อให้ได้สัญรูปของพื้นที่ซึ่งใช้พลังงานเพิ่มขึ้นในลำดับถัดไป

รูปที่ 4.13: หน้าจอขณะเล่นเกมส์

เกมส์จะสิ้นสุดลงเมื่อผู้เล่นไม่สามารถย้ายรูปใด ๆ เพิ่มเติมได้อีกเนื่องจากมีค่าเต็มทุกช่องแล้ว และด้านขวาของหน้าจอจะแสดงสัญรูปของพื้นที่ซึ่งผู้เล่นสามารถทำคะแนนได้สูงสุดพร้อมทั้งเฉลยข้อมูลการใช้พลังงานของพื้นที่ซึ่งผู้เล่นยังทำคะแนนได้ไม่ถึงดังรูปที่ 4.14

รูปที่ 4.14: หน้าจอเมื่อสิ้นสุดการเล่นเกมส์

แคนวาสแสดงข้อมูลประกาศจากทางภาควิชาวิศวกรรมไฟฟ้า (EE information) : ในส่วนนี้ใช้เพื่อการแสดงประกาศต่างๆในภาควิชาดังรูปที่ 4.15 ผู้วิจัยได้สร้างบัญชีผู้ใช้ของ drop-box ไว้เพื่อให้ผู้มีส่วนเกี่ยวข้องสามารถนำข้อมูลที่ต้องการประกาศบรรจุขึ้นในโฟลเดอร์ได้ผ่านโครงข่ายอินเทอร์เน็ต โดยมีบัญชีผู้ใช้ และรหัสผ่านดังตารางที่ 4.1 สำหรับไฟล์ที่รองรับในหน้าจอทำงานนี้ผู้วิจัยกำหนดให้ใช้ไฟล์ภาพซึ่งมีชนิดเป็นแบบ JPEG (Joint Photograhic's Experts Group) และ PNG (Portable Network Graphics) ที่มีขนาด 1920 x 930 จุดภาพ โดยข้อมูลภาพที่จะนำมาใช้ในการประกาศจะต้องบรรจุอยู่ในโฟลเดอร์ที่มีชื่อว่า "EE_information" ซึ่งมีบรรจะอยู่ภายในโฟลเดอร์ของทุกพื้นที่ติดตั้งได้แก่ CU_EE_fl12, CU_EE_fl13 และ CU_EE_HV ที่อยู่ภายใน dropbox โฟลเดอร์ตามลำดับ

ตารางที่ 4.1: บัญชีผู้ใช้ และรหัสผ่านของ dropbox เพื่อใช้บรรจุไฟล์ที่ต้องการประกาศ

username	eecubems@gmail.com
password	eecubems2014

รูปที่ 4.15: หน้าจอแสดงข้อมูลประกาศจากภาควิชา

แคนวาสแสดงการแจ้งเตือนการใช้พลังงานไฟฟ้า (alarm & alert) : ส่วนนี้เป็นส่วนสำหรับแสดงการแจ้งเตือนให้ทราบปริมาณการใช้พลังงานไฟฟ้าภายในภาควิชาเทียบกับขอบเขตที่ตั้งไว้ด้วยการเน้นสี (highlight) ซึ่งเป้าหมายที่จะแสดงผลเตือนคือ ระดับภาควิชาฯ (ตึก 4 และตึกไฟฟ้าฯ) ระดับตึก และระดับชั้น โดยวิธีการนำเสนอใช้การแสดงผลเป็นร้อยละของปริมาณการใช้พลังงานเทียบกับค่าขอบเขต และการเน้นสี(แดง เหลือง เขียว) เพื่อบอกระดับความเร่งด่วนของความต้องการการปรับปรุงการใช้พลังงาน

ในส่วนของการคำนวณเพื่อใช้ในการตั้งค่าสำหรับการเน้นสีนั้นผู้วิจัยได้ใช้นโยบายเดียวกันกับทางเว็บไซต์ของโครงการ CU-BEMS ซึ่งในขณะนี้ใช้การตั้งค่าอิสระ (manual) โดยผู้ดูแลระบบ ดังนี้

1. การคำนวณจะเริ่มตั้งแต่วันที่ 1 จนถึงสิ้นเดือน
2. เป้าหมายของพื้นที่ซึ่งจะแสดงผลเตือนเป็นไปตามตารางที่ 4.2
3. กำหนดโควต้าการใช้พลังงานโดยการจัดแบ่งโควต้าเป็นไปตามตารางที่ 4.2 ซึ่งถูกแบ่งตามชนิดของวัน (วันธรรมดา, วันเสาร์-อาทิตย์) รวมทั้งพื้นที่การใช้งานสำหรับการกำหนดค่าเป้าหมายดังกล่าวภายในแคนวาสนี้ ผู้วิจัยได้สร้างไฟล์ชื่อว่า tarketQouta.csv ซึ่งถูกบรรจุอยู่ใน dropbox โฟลเดอร์ไว้เพื่อให้ผู้มีส่วนเกี่ยวข้องสามารถปรับเปลี่ยนค่าเป้าหมายได้ในภายหลัง

ตารางที่ 4.2: โควต้าการใช้พลังงาน (kWh) ต่อวันสำหรับแต่ละพื้นที่ของโครงการ CU-BEMS ที่นำมาใช้ในโปรแกรมที่พัฒนาขึ้น

พื้นที่	วันธรรมดา	วันเสาร์	วันอาทิตย์
ภาควิชาวิศวกรรมไฟฟ้า (รวมทุกพื้นที่)	1035	871.795	435.897
ชั้น 2 อาคารภาควิชาวิศวกรรมไฟฟ้า	105.145	62.765	39.382
ชั้น 3 อาคารภาควิชาวิศวกรรมไฟฟ้า	60.463	40.312	20.155
ชั้น 4 อาคารภาควิชาวิศวกรรมไฟฟ้า	72.893	46.314	29.157
ชั้น 5 อาคารภาควิชาวิศวกรรมไฟฟ้า	49.841	33.231	16.615
ชั้น 12 อาคารเจริญวิศวกรรม	295.378	198.587	121.794
ชั้น 13 อาคารเจริญวิศวกรรม	401.438	267.290	195.690
อาคารปฏิบัติการไฟฟ้าแรงสูง	401.438	267.290	195.690

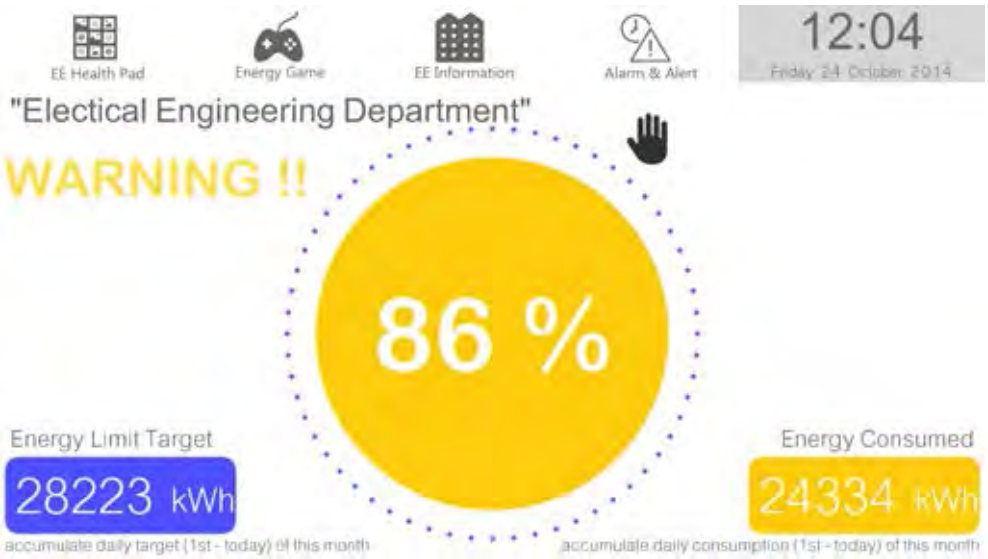
4. วิธีการกำหนดการเน้นสี เป็นดังนี้

   4.1. กำหนดให้
   actual คือ การใช้พลังงานจริงสะสมนับจากวันที่ 1 ของเดือน
   quota คือ โควต้าที่อนุญาตให้ใช้พลังงานสะสมนับจากวันที่ 1 ของเดือน
   ratio คือ ร้อยละของสัดส่วนการใช้พลังงานจริงสะสมเทียบกับโควต้าสะสม

   4.2. พิจารณาค่าการเน้นสี ณ วันที่ i
   actual(i) = ปริมานการใช้พลังงานสะสมจริงถึงเวลาปัจจุบันของวันที่ i
   quota(i) = quota(i-1)+ โควต้าที่อนุญาตให้ ณ วันที่ i โดยเริ่มตั้งค่าโควต้าใหม่ทุกเดือน นั่นคือ quota(0) = 0 kWh
   ratio = (actual(i)*100)/quota(i)

5. ใช้ ratio เป็นตัวกำหนดการเน้นสีตามเงื่อนไขที่กำหนด คือ
   น้อยกว่า หรือ เท่ากับ 75 กำหนดเป็น สีเขียว
   มากกว่า 75 แต่น้อยกว่า หรือ เท่ากับ 90 กำหนดเป็น สีเหลือง
   มากกว่า 90 กำหนดเป็น สีแดง

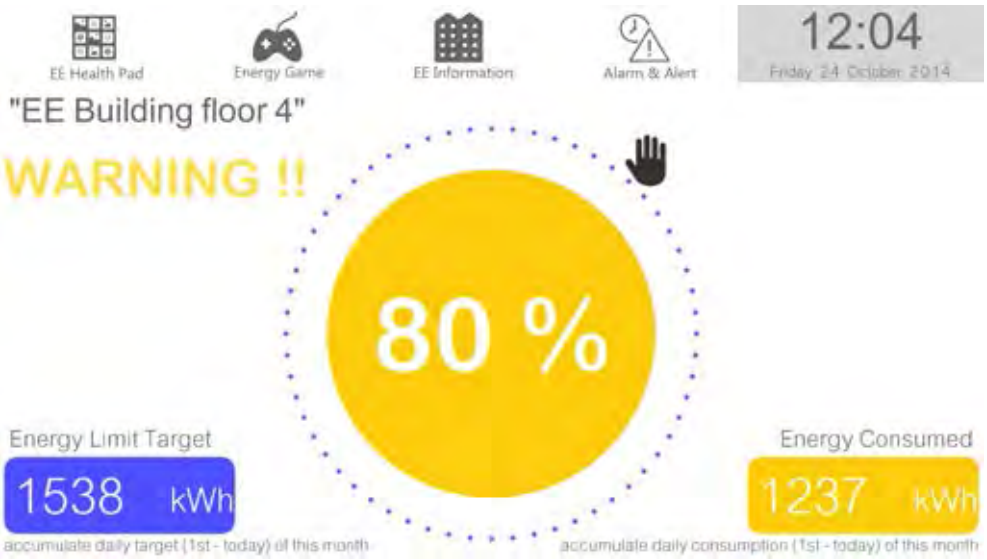
สำหรับส่วนแสดงผลหลักของแคนวาสนี้มีทั้งหมด 8 หน้าจอ ผู้ใช้งานสามารถใช้การโบกมือไปทางซ้ายหรือขวาเพื่อเลื่อนดูข้อมูลในบริเวณพื้นที่ก่อนหน้าหรือพื้นที่ถัดไปได้ โดยการแสดงผลในแต่ละหน้านั้นเป็นการนำค่า ratio ที่คำนวณได้มาใช้ในการวาดรูปวงกลมซึ่งมีขนาดของวงกลมแปรผันตามค่าของ ratio ที่ได้ นอกจากนั้นยังมีการวาดวงกลมเป็นเส้นประเพื่อแสดงตำแหน่งที่ ratio มีค่าเป็นร้อยละร้อย หรือเมื่อมีการใช้พลังงานเต็มจำนวนโควต้าที่ได้รับแล้ว นอกจากนั้นมีการเน้นสีของวงกลมตามวิธีที่กล่าวมาในข้างต้น ดังแสดงในรูปที่ 4.16 - 4.23 ตามลำดับ



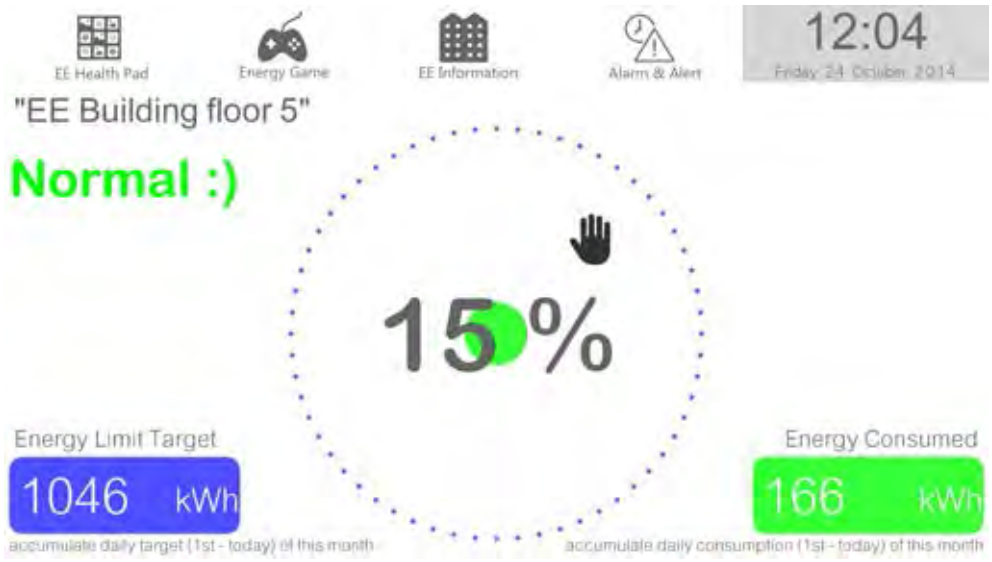
รูปที่ 4.16: หน้าจอแสดงข้อมูลการใช้พลังงานรวมทั้งภาควิชาคิดเป็นร้อยละเมื่อเทียบกับเป้าหมาย

รูปที่ 4.17: หน้าจอแสดงข้อมูลการใช้พลังงานของชั้น 2 อาคารภาควิชาวิศวกรรมไฟฟ้า คิดเป็นร้อยละเมื่อเทียบกับเป้าหมาย

รูปที่ 4.18: หน้าจอแสดงข้อมูลการใช้พลังงานของชั้น 3 อาคารภาควิชาวิศวกรรมไฟฟ้า คิดเป็นร้อยละเมื่อเทียบกับเป้าหมาย



รูปที่ 4.19: หน้าจอแสดงข้อมูลการใช้พลังงานของชั้น 4 อาคารภาควิชาวิศวกรรมไฟฟ้า คิดเป็นร้อยละเมื่อเทียบกับเป้าหมาย



รูปที่ 4.20: หน้าจอแสดงข้อมูลการใช้พลังงานของชั้น 5 อาคารภาควิชาวิศวกรรมไฟฟ้า คิดเป็นร้อยละเมื่อเทียบกับเป้าหมาย

รูปที่ 4.21: หน้าจอแสดงข้อมูลการใช้พลังงานของชั้น 12 อาคารเจริญวิศวกรรม คิดเป็นร้อยละเมื่อเทียบกับเป้าหมาย

รูปที่ 4.22: หน้าจอแสดงข้อมูลการใช้พลังงานของชั้น 13 อาคารเจริญวิศวกรรม คิดเป็นร้อยละเมื่อเทียบกับเป้าหมาย

รูปที่ 4.23: หน้าจอแสดงข้อมูลการใช้พลังงานของอาคารไฟฟ้าแรงสูง คิดเป็นร้อยละเมื่อเทียบกับเป้าหมาย

4.1.2 การควบคุมโปรแกรม : ใช้การแสดงท่าทางในการควบคุมการทำงาน

การแสดงท่าทางเพื่อควบคุมโปรแกรมที่พัฒนาขึ้น กำหนดให้ใช้มือข้างขวาเพียงข้างเดียวเนื่องจากผู้ใช้ทั่วไปถนัดการใช้มือข้างขวา และใช้การโบกมือหรือเลื่อนมือเท่านั้นการควบคุมโปรแกรมเพื่อให้ผู้ใช้งานสามารถใช้งานได้ง่าย ในส่วนของรูปแบบการใช้งานเป็นดังนี้

1. การเลือกเข้าใช้งานแคนวาสต่าง ๆ ทำได้โดยเลื่อนตัวชี้ตำแหน่งรูปมือไปวางบนสัญรูปของโปรแกรมย่อยนั้น 3 วินาที

2. การเลื่อนดูข้อมูลที่อยู่ภายในแคนวาสทำได้โดยใช้การโบกมือไปทางด้านซ้ายหรือขวาซึ่งเมื่อระยะของการโบกมือมากพอ ตัวชี้ตำแหน่งรูปมือจะเปลี่ยนจากสีดำเป็นสีขาว และโปรแกรมประยุกต์ย่อยจะแสดงข้อมูลในหน้าถัดไปจากนั้นผู้ใช้งานต้องรอ 1.5 วินาทีเพื่อที่จะควบคุมครั้งต่อไปหลังจากตัวชี้ตำแหน่งรูปมือเปลี่ยนจากสีขาวเป็นสีดำโดยระยะเวลา 1.5 วินาทีถูกกำหนดไว้เพื่อลดความผิดพลาดจากการดึงมือกลับหลังจากการโบกมือครั้งก่อนหน้า

4.1.3 การสื่อสารระหว่างโปรแกรมที่พัฒนาขึ้นกับหน่วยเก็บข้อมูลของโครงการ CU-BEMS : ใช้โพรโทคอล FETCH และ WRITE

งานวิจัยนี้ใช้โพรโทคอล FETCH ในการร้องขอข้อมูลเพื่อนำเสนอในส่วนแสดงผลหลักของโปรแกรมซึ่งมีตัวอย่างของโปรแกรมในส่วนของการร้องขอข้อมูลเป็นดังรูปที่ 4.24 และ 4.25 โดยในรูปที่ 4.24 เป็นการเตรียมข้อมูล xml ตามมาตรฐาน IEEE1888 เพื่อการร้องขอข้อมูล ในส่วนนี้ต้องระบุ point ID พร้อมทั้งช่วงเวลาที่เราต้องการร้องขอข้อมูลด้วย จากนั้นรูปที่ 4.25 เป็นส่วนของการส่งข้อมูลการร้องขอไปยังหน่วยเก็บข้อมูลของโครงการ CU-BEMS

```
void ieee1888_fetch() {
  try {
    String xmldata = "<?xml version=\"1.0\""+
      "encoding=\"UTF-8\"?>" +
      "<soapenv:Envelope " +
      "xmlns:soapenv="+
      "\"http://schemas.xmlsoap.org/soap/envelope/\">"+
      "<soapenv:Body>"+
      "<ns2:queryRQ xmlns:ns2=\"http://soap.fiap.org/\">"+
      "<transport xmlns=\"http://gutp.jp/fiap/2009/11/\">"+
      "<header>"+
      "<query id=\"9eed9de4-1c48-4b08-a41d-dac067fc1c0d\""+
      "type=\"storage\">"+
      pointID[0]+pointID[1]+pointID[2]+pointID[3]+pointID[4]+
      pointID[5]+pointID[6]+pointID[7]+pointID[8]+
      "</query></header></transport></ns2:queryRQ>"+
      "</soapenv:Body></soapenv:Envelope>";

    //Create socket//
    String hostname = "161.200.90.122"; int port = 80;
    InetAddress  addr = InetAddress.getByName(hostname);
    Socket sock = new Socket(addr, port);
```

รูปที่ 4.24: โปรแกรมในส่วนของการร้องขอข้อมูลโดยใช้โพรโทคอล FETCH

```
    //Send header//
    String path = "/axis2/services/FIAPStorage";
    BufferedWriter wr =
    new BufferedWriter(new OutputStreamWriter(sock.getOutputStream(), "UTF-8"));
    wr.write("POST " + path + " HTTP/1.0\r\n");
    wr.write("Host: 161.200.90.122\r\n");
    wr.write("Content-Length: " + xmldata.length() + "\r\n");
    wr.write("Content-Type: text/xml; charset=\"utf-8\"\r\n");
    wr.write("SOAPAction: http://soap.fiap.org/query\r\n");
    wr.write("\r\n");

    //Send data//
    wr.write(xmldata);
    wr.flush();

    // Response//
    BufferedReader rd =
    new BufferedReader(new InputStreamReader(sock.getInputStream()));
    String line;
    while ( (line = rd.readLine ()) != null) {
      fetch_res = fetch_res+line;
    }
  }catch (Exception e) {
    e.printStackTrace();
  }
```

รูปที่ 4.25: โปรแกรมในส่วนของการร้องขอข้อมูลโดยใช้โพรโทคอล FETCH (ต่อ)

โปรแกรมข้างต้นนี้เป็นการร้องขอข้อมูลจาก point ID 9 ค่า ที่ได้ประกาศเป็นตัวแปรส่วนกลางในส่วนบนสุดของโปรแกรมหลักดังรูปที่ 4.26 และข้อมูลที่ตอบกลับมาจากหน่วยเก็บข้อมูลจะถูกนำมาใช้แสดงผลในส่วนของโปรแกรม EE health pad รูปแบบโปรแกรมเพื่อการร้องขอข้อมูลด้วยโพรโทคอล FETCH ในส่วนอื่น ๆ นั้นจะใช้รูปแบบเดียวกันเปลี่ยนเพียง point ID ของข้อมูลที่ต้องการร้องขอ

```
String [] pointID = {
  "<key id=\"bems.ee.eng.chula.ac.th/ee_health_pad/"
  +"co2_reduction\" attrName=\"time\" select=\"maximum\" />",
  "<key id=\"bems.ee.eng.chula.ac.th/ee_health_pad/"
  +"coal_reduction\" attrName=\"time\" select=\"maximum\" />",
  "<key id=\"bems.ee.eng.chula.ac.th/ee_health_pad/"
  +"consumption\" attrName=\"time\" select=\"maximum\" />",
  "<key id=\"bems.ee.eng.chula.ac.th/ee_health_pad/"
  +"electricity_saved\" attrName=\"time\" select=\"maximum\" />",
  "<key id=\"bems.ee.eng.chula.ac.th/ee_health_pad/"
  +"energy_saved\" attrName=\"time\" select=\"maximum\" />",
  "<key id=\"bems.ee.eng.chula.ac.th/ee_health_pad/"
  +"load_factor\" attrName=\"time\" select=\"maximum\" />",
  "<key id=\"bems.ee.eng.chula.ac.th/ee_health_pad/"
  +"peak_saved\" attrName=\"time\" select=\"maximum\" />",
  "<key id=\"bems.ee.eng.chula.ac.th/ee_health_pad/"
  +"renewable_energy\" attrName=\"time\" select=\"maximum\" />",
  "<key id=\"bems.ee.eng.chula.ac.th/ee_health_pad/"
  +"tree_saved\" attrName=\"time\" select=\"maximum\" />"
};
```

รูปที่ 4.26: point ID ที่ใช้ในการร้องขอข้อมูล

วิทยานิพนธ์นี้ยังมีการใช้โพรโทคอล WRITE เพื่อส่งข้อมูลพฤติกรรมของผู้ใช้งานไปเก็บไว้ในหน่วยเก็บข้อมูลของโครงการ CU-BEMS สำหรับการประเมินการทำงานของโปรแกรมที่พัฒนาขึ้นซึ่งโปรแกรมที่ใช้ส่งข้อมูลแสดงไว้ในรูปที่ 4.27- 4.29 โดยในรูปที่ 4.27 เป็นส่วนของการเตรียมรูปแบบของตราเวลาให้ตรงตามมาตรฐาน IEEE1888 เพื่อใช้ในการส่งไปพร้อมกับข้อมูลที่เราต้องการใช้ในการประเมิน 3 ข้อมูล ได้แก่ *noUser*, *canvas_number* และ *noSwipe* ดังรูปที่ 4.28 ซึ่งเป็นข้อมูลของจำนวนครั้งที่ระบบถูกกระตุ้นให้เข้าสู่แคนวาสปริยายจากคนที่เดินผ่าน, หมายเลขของแคนวาสที่ถูกเรียกใช้งาน และจำนวนการโบกมือของผู้ใช้งานในการเข้าใช้แคนวาสครั้งก่อนหน้าตามลำดับ จากนั้นในรูปที่ 4.25 จะเป็นส่วนของการส่งข้อมูลออกไปเก็บยังหน่วยเก็บข้อมูลของโครงการ CU-BEMS

```
void ieee1888_write() {

  write_year  = ""+year();   // 2003, 2004, 2005, etc.
  if (month() < 10){write_month = "0"+month();}
  else {write_month = ""+month();}
  if (day() < 10){write_day = "0"+day();}
  else {write_day = ""+day();}
  if (hour() < 10) {write_hour = "0"+hour();}
  else {write_hour = ""+hour();}
  if (minute() < 10) {write_min = "0"+minute();}
  else {write_min = ""+minute();}
  if (second() < 10) {write_sec = "0"+second();}
  else {write_sec = ""+second();}
```

รูปที่ 4.27: โปรแกรมในส่วนของการเขียนข้อมูลโดยใช้โพรโทคอล WRITE

```
try {String xmldata = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>"+
    "<soapenv:Envelope " +
    "xmlns:soapenv="+
    "\"http://schemas.xmlsoap.org/soap/envelope/\">"+
    "<soapenv:Body>"+
    "<ns2:dataRQ xmlns:ns2=\"http://soap.fiap.org/\">"+
    "<transport xmlns=\"http://gutp.jp/fiap/2009/11/\">"+
    "<body>"+
    "<pointSet id=\"http://bems.ee.eng.chula.ac.th/eng4/"+
    "fl13/corridor/elevatorfront/kinect\">"+
    "<point id=\"http://bems.ee.eng.chula.ac.th/eng4/"+
    "fl13/corridor/elevatorfront/kinect/num_user\">"+
    "<value time=\""+write_year+"-"+write_month+"-"+write_day+
    "T"+write_hour+":"+write_min+":"+write_sec+".000+07:00\">"+
    noUser+"</value>"+"</point>"+
    "<point id=\"http://bems.ee.eng.chula.ac.th/eng4/"+
    "fl13/corridor/elevatorfront/kinect/canvas_num\">"+
    "<value time=\""+write_year+"-"+write_month+"-"+write_day+
    "T"+write_hour+":"+write_min+":"+write_sec+".000+07:00\">"+
    "canvas"+canvas_number+"</value>"+"</point>"+
    "<point id=\"http://bems.ee.eng.chula.ac.th/eng4/"+
    "fl13/corridor/elevatorfront/kinect/cumm_swipe\">"+
    "<value time=\""+write_year+"-"+write_month+"-"+write_day+
    "T"+write_hour+":"+write_min+":"+write_sec+".000+07:00\">"+
    "swipe"+noSwipe+"</value>"+"</point>"+
    "</pointSet>"+"</body>"+
    "</transport></ns2:dataRQ>"+
    "</soapenv:Body></soapenv:Envelope>";
```

รูปที่ 4.28: โปรแกรมในส่วนของการเขียนข้อมูลโดยใช้โพรโทคอล WRITE (ต่อ)

```
    //Create socket
    String hostname = "161.200.90.122"; int port = 80;
    InetAddress  addr = InetAddress.getByName(hostname);
    Socket sock = new Socket(addr, port);

    //Send header
    String path = "/axis2/services/FIAPStorage";
    BufferedWriter wr =
    new BufferedWriter(new OutputStreamWriter(sock.getOutputStream(),"UTF-8"));
    wr.write("POST " + path + " HTTP/1.0\r\n");
    wr.write("Host: 161.200.90.122\r\n");
    wr.write("Content-Length: " + xmldata.length() + "\r\n");
    wr.write("Content-Type: text/xml; charset=\"utf-8\"\r\n");
    wr.write("SOAPAction: http://soap.fiap.org/data\r\n");
    wr.write("\r\n");

    //Send data
    wr.write(xmldata); wr.flush();
    // Response
    BufferedReader rd =
    new BufferedReader(new InputStreamReader(sock.getInputStream()));
    String line;
    while ( (line = rd.readLine ()) != null) {
    }
  } catch (Exception e) {
    e.printStackTrace();
  }
}
```

รูปที่ 4.29: โปรแกรมในส่วนของการเขียนข้อมูลโดยใช้โพรโทคอล WRITE (ต่อ)

4.2 การทดสอบการใช้งานจริงในโครงการ CU-BEMS

ในส่วนนี้จะเป็นการกล่าวถึงการนำระบบที่พัฒนาขึ้นมาใช้งานจริงในโครงการ CU-BEMS โดยระบบที่พัฒนาขึ้นนี้ประกอบไปด้วยซอฟต์แวร์ซึ่งมีรายละเอียดเป็นดังตารางที่ 4.3 ซอฟต์แวร์ทั้งหมดจะถูกติดตั้งลงบนคอมพิวเตอร์ที่มีรายละเอียดดังแสดงในตารางดังตารางที่ 4.4 ซึ่งเป็นตารางแสดงรายละเอียดของฮาร์ดแวร์ทั้งหมดที่ใช้ในระบบที่พัฒนาขึ้น

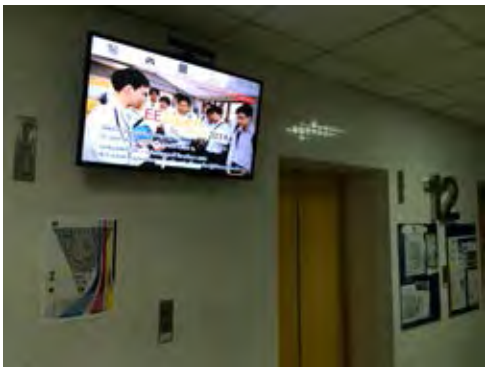
ตารางที่ 4.3: รายละเอียดของซอฟต์แวร์ที่ใช้ในการพัฒนาโปรแกรมสร้างมโนภาพ

software	**description**
operating system	Windows 7 Ultimate 64-bit
programming language	processing version 2.0
software development kit	kinect SDK
libraries	SimpleOpenNI

ตารางที่ 4.4: รายละเอียดของฮาร์ดแวร์ที่ใช้ในการพัฒนาโปรแกรมสร้างมโนภาพ

hardware	**description**
computer	mini PC intel NUC intel core i5-4250U 1.30 GHz processor 4 GB RAM
gesture sensor	kinect for xbox360
display	42-inch full HD (1920 x 1080)

ชุดอุปกรณ์ของระบบที่พัฒนาขึ้นตามรายละเอียดในข้างต้นทั้งหมดจะถูกติดตั้งไว้ในบริเวณภาควิชาวิศวกรรมไฟฟ้ารวมทั้งสิ้น 3 จุด คือ บริเวณโถงลิฟท์ชั้น 12 และชั้น 13 ของอาคารเจริญวิศวกรรม รวมทั้งบริเวณหน้าทางเข้าอาคารปฏิบัติการวิศวกรรมไฟฟ้าแรงสูง โดยในรูปที่ 4.30(a) และ 4.30(b) เป็นรูปแสดงจุดติดตั้งเพื่อการใช้งานจริงบริเวณโถงลิฟท์ชั้น 12 และชั้น 13 ของอาคารเจริญวิศวกรรมตามลำดับ ในส่วนของพื้นที่ติดตั้งบริเวณหน้าทางเข้าอาคารปฏิบัติการวิศวกรรมไฟฟ้าแรงสูงแสดงในรูปที่ 4.30(c) และ 4.30(d)



(a) ชั้น 12 อาคารเจริญวิศวกรรม

(b) ชั้น 13 อาคารเจริญวิศวกรรม

(c) อาคารปฏิบัติการไฟฟ้าแรงสูง

(d) บริเวณอาคารปฏิบัติการไฟฟ้าแรงสูง

รูปที่ 4.30: จุดติดตั้งชุดอุปกรณ์ของระบบที่พัฒนาขึ้น

4.2.1 การทดสอบเพื่อประเมินพฤติกรรมของผู้เข้าใช้งานระบบ

จากที่ได้กล่าวไว้ในข้างต้นว่าระบบที่พัฒนาขึ้นมีคุณสมบัติที่สามารถเก็บข้อมูลพฤติกรรมของผู้ใช้งานแต่ละคนโดยอัตโนมัติ ผู้วิจัยจึงทดสอบเก็บข้อมูลจากการใช้งานจริงทั้ง 3 จุดติดตั้งเป็นเวลา 1 สัปดาห์เพื่อนำข้อมูลที่ได้มาใช้ในการประเมินการทำงานของโปรแกรม ข้อมูลพฤติกรรมของผู้ใช้งานที่นำมาใช้มีทั้งหมด 3 ชนิด ดังนี้

- จำนวนการโบกมือของผู้ใช้งานในการเข้าใช้แคนวาสครั้งก่อนหน้า (num_swipe) กำหนดให้นับจำนวนเมื่อผู้ใช้ควบคุมโปรแกรมด้วยการโบกมือสำเร็จ (เมื่อตัวชี้ตำแหน่งมือเปลี่ยนจากสีดำเป็นสีขาว)
- หมายเลขของแคนวาสที่ถูกเรียกใช้งาน (canvas_num)
- จำนวนครั้งที่ระบบถูกกระตุ้นให้เข้าสู่แคนวาสปริยายจากคนที่เดินผ่าน (num_user) กำหนดให้นับจำนวนเมื่อหมายเลขแคนวาสเปลี่ยนจาก 0 เป็นหมายเลขของแคนวาสปริยาย

ผู้วิจัยกำหนดให้ข้อมูลพฤติกรรมของผู้ใช้งานถูกส่งมาเก็บไว้ในหน่วยเก็บข้อมูลของโครงการ CU-BEMS พร้อมกันทั้ง 3 ค่าในทุกครั้งที่มีการเปลี่ยนหมายเลขแคนวาสโดยมี point ID ที่ใช้สำหรับเก็บข้อมูลของแต่ละพื้นที่ดังนี้

ชั้น 12 อาคารเจริญวิศวกรรม

- http://bems.ee.eng.chula.ac.th/eng4/fl12/corridor/elevatorfront/kinect/num_swipe
- http://bems.ee.eng.chula.ac.th/eng4/fl12/corridor/elevatorfront/kinect/canvas_num
- http://bems.ee.eng.chula.ac.th/eng4/fl12/corridor/elevatorfront/kinect/num_user

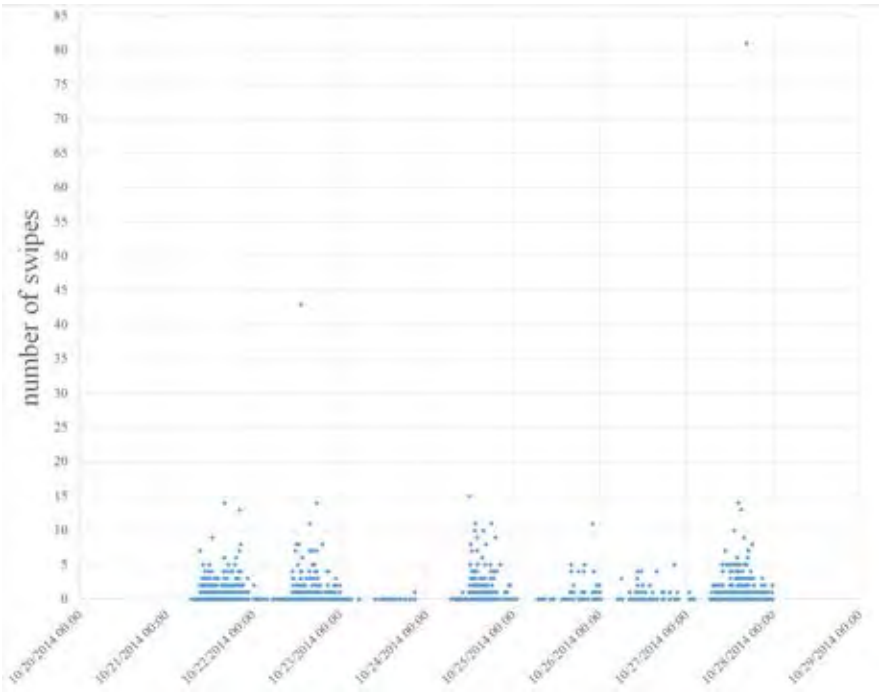
ชั้น 13 อาคารเจริญวิศวกรรม

- http://bems.ee.eng.chula.ac.th/eng4/fl13/corridor/elevatorfront/kinect/num_swipe
- http://bems.ee.eng.chula.ac.th/eng4/fl13/corridor/elevatorfront/kinect/canvas_num
- http://bems.ee.eng.chula.ac.th/eng4/fl13/corridor/elevatorfront/kinect/num_user

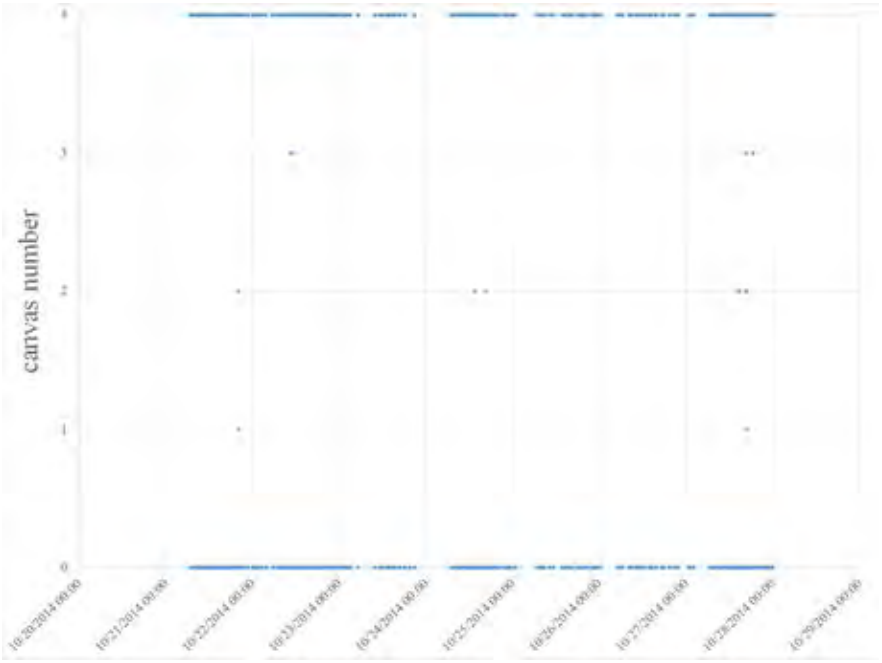
อาคารปฏิบัติการไฟฟ้าแรงสูง

- http://bems.ee.eng.chula.ac.th/highvoltage/kinect/num_swipe
- http://bems.ee.eng.chula.ac.th/highvoltage/kinect/canvas_num
- http://bems.ee.eng.chula.ac.th/highvoltage/kinect/num_user

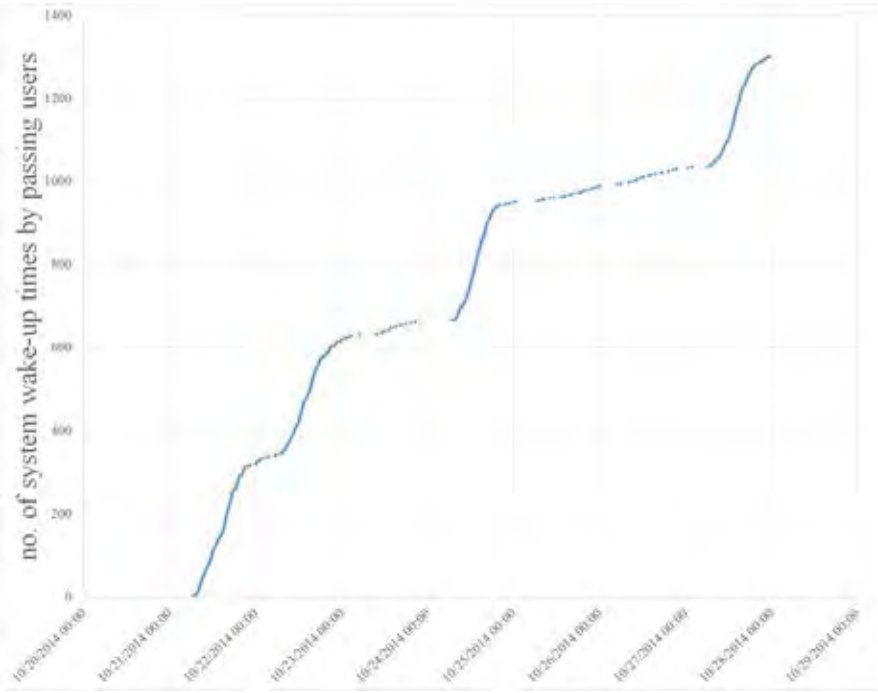
หลังจากเก็บข้อมูลเป็นเวลา 1 สัปดาห์จะได้ข้อมูลดิบของจำนวนการโบกมือของผู้ใช้งานในการเข้าใช้แคนวาสครั้งก่อนหน้า, หมายเลขของแคนวาสที่ถูกเรียกใช้งาน และจำนวนครั้งที่ระบบถูกกระตุ้นให้เข้าสู่แคนวาสปริยายจากคนที่เดินผ่านของพื้นที่ติดตั้งแต่ละจุดซึ่งสามารถนำมาทำเป็นแผนภาพกระจาย (scatter plot) ได้ดังรูปที่ 4.31 จนถึง 4.39 ตามลำดับ



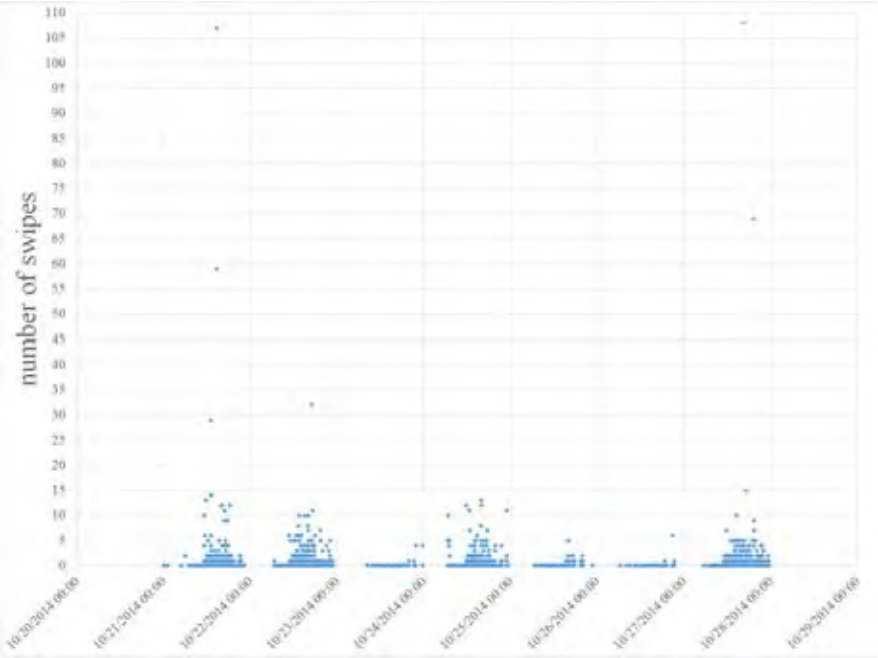
รูปที่ 4.31: ข้อมูลดิบของจำนวนการโบกมือของผู้ใช้งานในการเข้าใช้แคนวาสครั้งก่อนหน้าบริเวณชั้น 12 อาคารเจริญวิศวกรรม



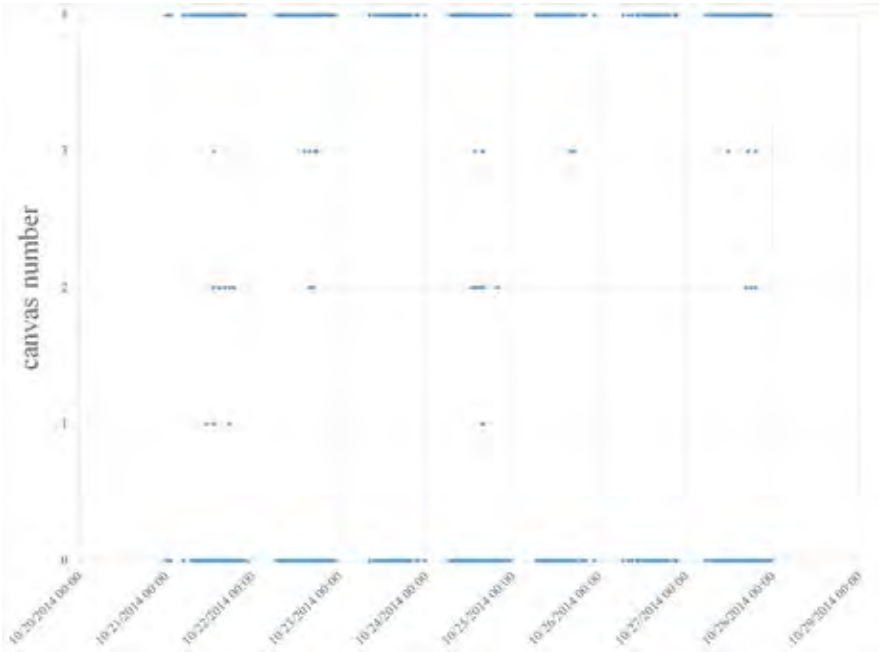
รูปที่ 4.32: ข้อมูลดิบของหมายเลขแคนวาสที่ถูกเรียกใช้งานบริเวณชั้น 12 อาคารเจริญวิศวกรรม



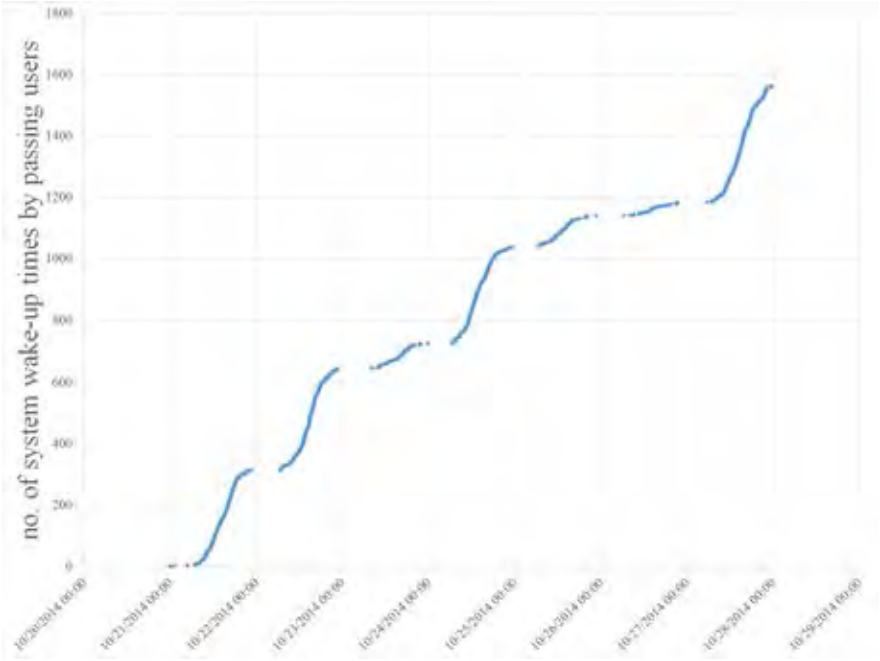
รูปที่ 4.33: ข้อมูลดิบของจำนวนครั้งที่ระบบถูกกระตุ้นให้เข้าสู่แคนวาสปริยายจากคนที่เดินผ่านบริเวณชั้น 12 อาคารเจริญวิศวกรรม



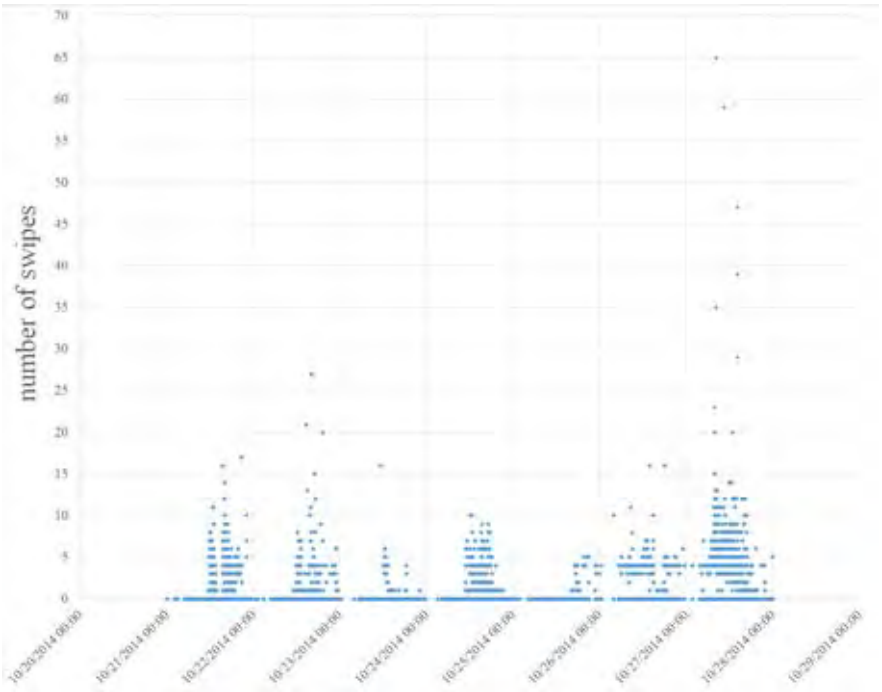
รูปที่ 4.34: ข้อมูลดิบของจำนวนการโบกมือของผู้ใช้งานในการเข้าใช้แคนวาสครั้งก่อนหน้าบริเวณชั้น 13 อาคารเจริญวิศวกรรม



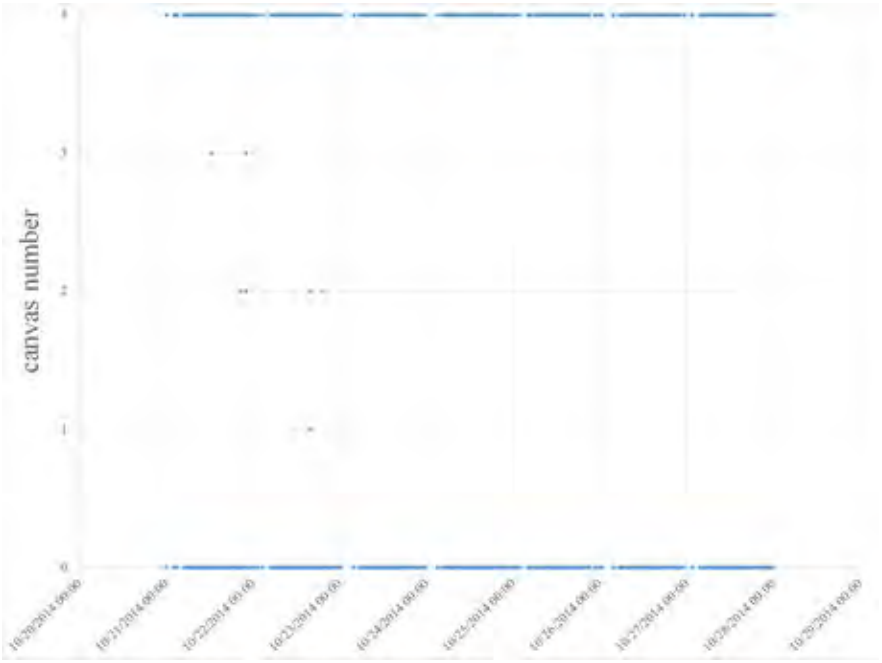
รูปที่ 4.35: ข้อมูลดิบของหมายเลขแคนวาสที่ถูกเรียกใช้งานบริเวณชั้น 13 อาคารเจริญวิศวกรรม



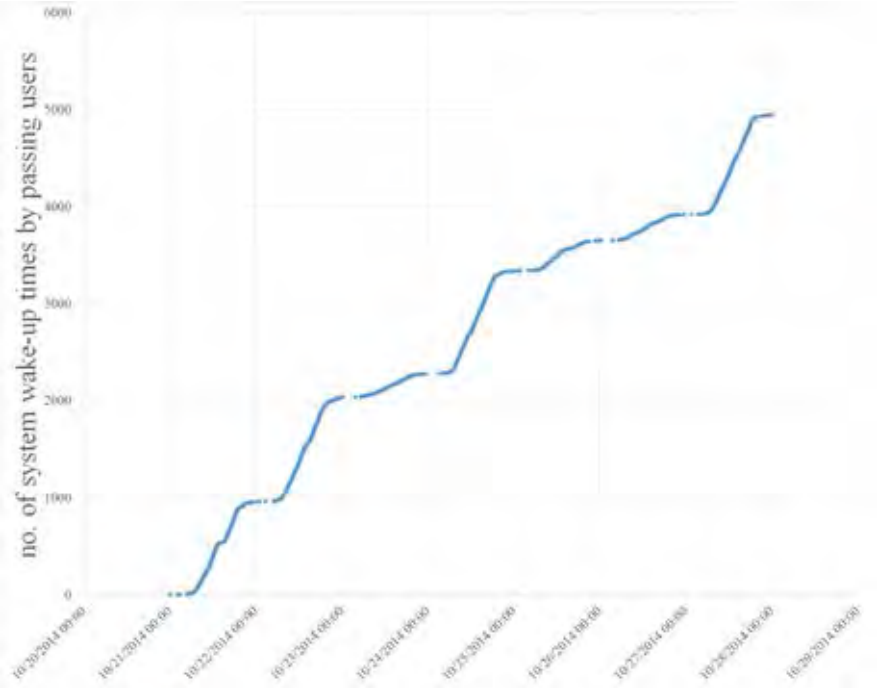
รูปที่ 4.36: ข้อมูลดิบของจำนวนครั้งที่ระบบถูกกระตุ้นให้เข้าสู่แคนวาสปริยายจากคนที่เดินผ่านบริเวณชั้น 13 อาคารเจริญวิศวกรรม



รูปที่ 4.37: ข้อมูลดิบของจำนวนการโบกมือของผู้ใช้งานในการเข้าใช้แคนวาสครั้งก่อนหน้าบริเวณอาคารปฏิบัติการไฟฟ้าแรงสูง



รูปที่ 4.38: ข้อมูลดิบของหมายเลขแคนวาสที่ถูกเรียกใช้งานบริเวณอาคารปฏิบัติการไฟฟ้าแรงสูง



รูปที่ 4.39: ข้อมูลดิบของจำนวนครั้งที่ระบบถูกกระตุ้นให้เข้าสู่แคนวาสปริยายจากคนที่เดินผ่านบริเวณอาคารปฏิบัติการไฟฟ้าแรงสูง

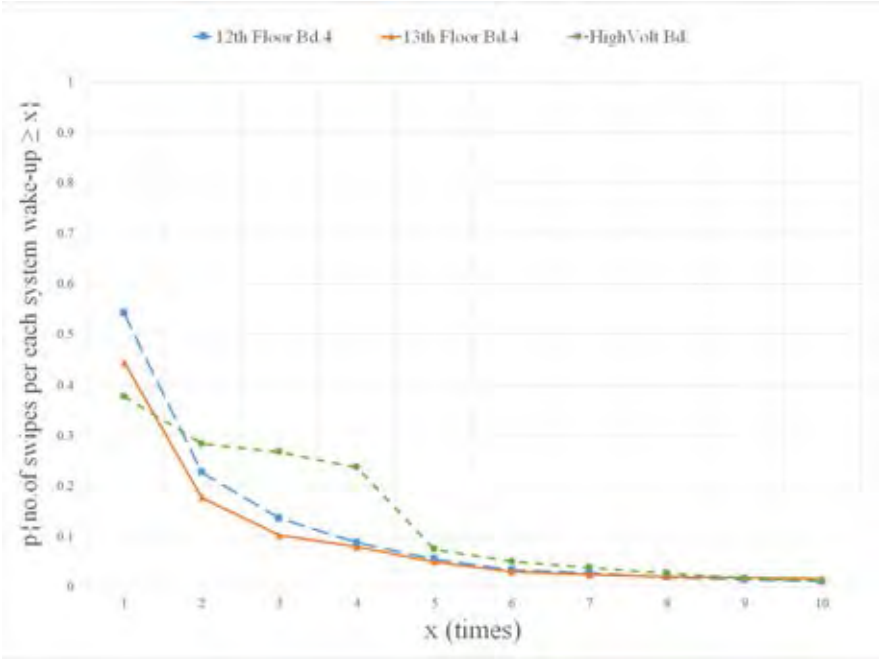
4.2.2 กรณีศึกษา 1 : เปรียบเทียบพฤติกรรมผู้ใช้งานในพื้นที่ติดตั้งแต่ละจุด

เมื่อนำข้อมูลดิบของจำนวนครั้งที่ระบบถูกกระตุ้นให้เข้าสู่แคนวาสปริยายจากคนที่เดินผ่านมาทำการศึกษาพบว่าภายใน 1 สัปดาห์ จุดที่ระบบถูกกระตุ้นให้เข้าสู่แคนวาสปริยายจากคนที่เดินผ่านมากที่สุดคือบริเวณหน้าทางเข้าอาคารปฏิบัติการไฟฟ้าแรงสูง ชั้น 13 อาคารเจริญวิศวกรรม และชั้น 12 อาคารเจริญวิศวกรรม ตามลำดับ ซึ่งมีข้อมูลแยกเป็นวันธรรมดา และวันเสาร์อาทิตย์ ดังแสดงในตารางที่ 4.5

ตารางที่ 4.5: จำนวนครั้งที่ระบบถูกกระตุ้นให้เข้าสู่แคนวาสปริยายจากคนที่เดินผ่านใน 1 สัปดาห์

	จำนวนครั้ง		
	ชั้น 12 อาคาร 4	ชั้น 13 อาคาร 4	อาคารปฏิบัติการไฟฟ้าแรงสูง
วันธรรมดา	1218	1418	4367
วันเสาร์-อาทิตย์	83	143	583

จากนั้นผู้วิจัยนำข้อมูลจากตารางที่ 4.5 ร่วมกับข้อมูลจำนวนการโบกมือของผู้ใช้งานในการเข้าใช้แคนวาสครั้งก่อนหน้าเพื่อหาว่าหลังจากที่ระบบถูกกระตุ้นจากคนที่เดินผ่านนั้นระบบได้รับความสนใจหรือไม่ โดยการหาความถี่ของคนที่ผ่านมาแล้วโบกมือ 1 ครั้ง, 2 ครั้ง ไปจนถึงจำนวนครั้งที่มากที่สุดที่มีการโบกมือจากนั้นนำมาเทียบกับจำนวนครั้งที่ระบบถูกกระตุ้นให้เข้าสู่แคนวาสปริยายจากคนที่เดินผ่านทั้งหมดในแต่ละพื้นที่



รูปที่ 4.40: ร้อยละของผู้ใช้งานที่มีการโต้ตอบกับระบบที่พัฒนาขึ้นในแต่ละจุดติดตั้ง

ผลจากการศึกษาข้อมูลดังกล่าวเป็นดังรูปที่ 4.40 ถ้าเริ่มนับจากคนที่ผ่านมาแล้วโบกมือ 1 ครั้งขึ้นไป บริเวณชั้น 12 อาคารเจริญวิศวกรรม (ตึก4) ระบบได้รับความสนใจสูงที่สุดโดยคิดเป็นร้อยละ 54.2 บริเวณชั้น 13 อาคารเจริญวิศวกรรม (ตึก4) ระบบได้รับความสนใจคิดเป็นจำนวนร้อยละ 44.6 และบริเวณอาคารปฏิบัติการไฟฟ้าแรงสูงระบบได้รับความสนใจคิดเป็นจำนวนร้อยละ 37.8 แต่ถ้าเริ่มนับจากคนที่ผ่านมาแล้วโบกมือ 2 ครั้งขึ้นไปบริเวณอาคารปฏิบัติการไฟฟ้าแรงสูงระบบได้รับความสนใจสูงสุดตามด้วยบริเวณชั้น 12 อาคารเจริญวิศวกรรม (ตึก4) และบริเวณชั้น 13 อาคารเจริญวิศวกรรม (ตึก4) ตามลำดับ แต่จากการตรวจสอบพบว่าบริเวณอาคารปฏิบัติการไฟฟ้าแรงสูงมีค่าร้อยละมากกว่าความเป็นจริงเนื่องจากตัวรับรู้ท่าทางถูกรบกวนโดย 2 ปัจจัยหลัก ดังนี้

1. กระจกติดฟิล์มกรองแสง : จากรูปที่ 4.30(c) จะเห็นว่าตัวรับรู้ท่าทางถูกติดตั้งอยู่หลังกระจกซึ่งเป็นกระจกติดฟิล์มกรองแสงซึ่งทำให้ระยะการตรวจจับของตัวรับรู้สั้นลงรวมทั้งทำให้เกิดความผิดพลาดในการติดตามโครงกระดูกของผู้ใช้งานส่งผลให้การควบคุมการทำงานของโปรแกรมด้วยตัวรับรู้ท่าทางเกิดการรบกวน

2. แสงแดด : เนื่องจากตัวรับรู้ท่าทางตรวจจับ และการติดตามโครงกระดูกของผู้ใช้โดยการทำงานร่วมกันระหว่างตัวจ่ายแส่ง และตัวรับรู้อินฟาเรด จากรูปที่ 4.30(d) จะเห็นว่าบริเวณการใช้งานหน้าอาคารปฏิบัติการไฟฟ้าแรงสูงเป็นพื้นที่กลางแจ้งซึ่งในช่วงกลางวันมีแสงแดดส่องถึง รังสีอินฟาเรดในแสงแดดจะรบกวนการทำงานของตัวรับรู้ท่าทางในการติดตามโครงกระดูกของผู้ใช้งานทำให้การควบคุมการทำงานของโปรแกรมเกิดความคลาดเคลื่อน

ด้วยปัจจัยดังที่กล่าวมานั้นผู้วิจัยมีความเห็นว่าจำเป็นต้องปรับปรุงจุดติดตั้งของตัวรับรู้ท่าทางให้อยู่ด้านนอกกระจกเพื่อเพิ่มเสถียรภาพในการตรวจจับของตัวรับรู้ นอกจากนั้นในช่วงเวลากลางวันขอเสนอให้พัฒนาโปรแกรมให้มีรูปแบบการแสดงผลอัตโนมัติเมื่อมีการตรวจพบผู้ใช้งานเพื่อหลีกเลี่ยงปัญหาของการรบกวนจากรังสีอินฟราเรดในแสงแดด

4.2.3 กรณีศึกษา 2 : เปรียบเทียบพฤติกรรมของผู้ใช้งานในแต่ละแคนวาส (เมื่อมีการตั้งแคนวาสปริยาย)

เมื่อนำข้อมูลหมายเลขแคนวาสที่ถูกเรียกใช้งานใน 1 สัปดาห์ซึ่งมีข้อมูลดังแสดงในตารางที่ 4.6 มาทำการศึกษาพฤติกรรมการของผู้ใช้งานพบว่าแคนวาส alarm & alert มีการเรียกใช้งานมากที่สุดในทุก ๆ พื้นที่ติดตั้งดังตารางที่ 4.7 เนื่องจากถูกตั้งค่าไว้เป็นแคนวาสปริยายของระบบดังนั้นทุกครั้งที่คนเดินผ่านจะถูกตั้งค่าให้เปลี่ยนจากแคนวาสแรกสุดซึ่งทำหน้าที่เป็น screen saver มายังแคนวาสซึ่งถูกตั้งค่าไว้ให้ค่าปริยายโดยอัตโนมัติทำให้ผู้ใช้งานมีโอกาสที่จะเห็นข้อมูลในแคนวาสปริยายมากที่สุด นอกจากนั้นจะเห็นว่าผู้ใช้งานมีความสนใจที่จะเรียกใช้งานในแคนวาสอื่น ๆ นอกเหนือจากแคนวาสปริยายที่ตั้งไว้น้อยมาก

ตารางที่ 4.6: รายละเอียดของหมายเลขแคนวาส

number	description of canvas
0	screen saver
1	EE health pad
2	energy game
3	EE information
4	alarm & alert

ตารางที่ 4.7: จำนวนครั้งที่แคนวาสถูกเรียกใช้งานเมื่อมีการตั้งแคนวาสปริยายไว้ที่หน้า alarm & alert

	ชั้น 12 อาคาร 4	ชั้น 13 อาคาร 4	อาคารปฏิบัติการไฟฟ้าแรงสูง
EE health pad	2	5	1
energy game	7	22	5
EE information	4	14	2
alarm & alert	1322	1583	4975

ผู้วิจัยได้นำข้อมูลจากตางรางที่ 4.7 มาศึกษาเพิ่มเติมในส่วนของการเรียกใช้งานในแคนวาส alarm & alert เท่านั้นเนื่องจากเป็นแคนวาสที่ถูกตั้งไว้เป็นแคนวาสปริยายดังนั้นข้อมูลที่ได้จึงไม่สามารถสรุปได้ว่าผู้ใช้งานสนใจที่จะเรียกใช้แคนวาส alarm & alert มากน้อยเพียงใด และสาเหตุที่ไม่วิเคราะห์ผู้ใช้งานที่สนใจเรียกใช้แคนวาสอื่นเพิ่มเติ่มเนื่องจาก การที่ผู้ใช้งานจะสามารถเรียกใช้แคนวาสอื่น ๆ นอกจากแคนวาสปริยายนั้นจำเป็นต้องเลื่อนตัวชี้ตำแหน่งรูปมือไปวางบนสัญรูปของแคนวาสนั้น 3 วินาที ดังนั้นผู้วิจัยจึงถือว่าข้อมูลในส่วนของการเรียกใช้แคนวาสอื่นนอกเหนือจากแคนวาสปริยายนั้นผู้ใช้งานจงใจที่จะเรียกใช้งาน และด้วยเหตุนี้ผู้วิจัยจึงขอกำหนดให้ผู้ใช้งานที่สนใจเรียกใช้แคนวาส alarm & alert คือ ผู้ใช้งานที่มีการโบกมือตั้งแต่ 2 ครั้งขึ้นไปในการเรียกใช้แคนวาส alarm & alert แต่ละครั้ง ผลที่ได้พบว่าแคนวาส alarm & alert ยังคงเป็นแคนวาสมีการเรียกใช้งานมากที่สุดในทุก ๆ พื้นที่ติดตั้งเช่นเดิมดังข้อมูลในตารางที่ 4.8

ตารางที่ 4.8: จำนวนครั้งที่แคนวาสถูกเรียกใช้งานเมื่อมีการตั้งแคนวาสปริยายไว้ที่หน้า alarm & alert และกำหนดให้ผู้ใช้งานที่สนใจเรียกใช้แคนวาส alarm & alert คือผู้ใช้งานที่มีการโบกมือตั้งแต่ 2 ครั้งขึ้นไปในการเรียกใช้แต่ละครั้ง

	ชั้น 12 อาคาร 4	ชั้น 13 อาคาร 4	อาคารปฏิบัติการไฟฟ้าแรงสูง
EE health pad	2	5	1
energy game	7	22	5
EE information	4	14	2
alarm & alert	286	256	1405

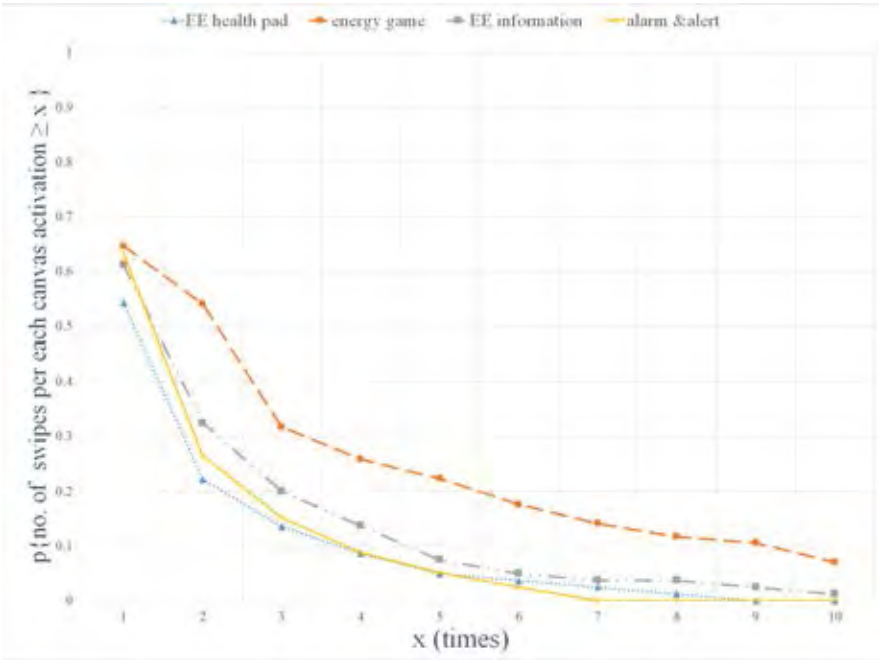
ผลจากข้อมูลข้างต้นผู้วิจัยจึงขอแสดงความเห็นว่าหากต้องการนำเสนอข้อมูลใดเป็นข้อมูลหลักให้กับผู้ใช้งานควรตั้งค่าไว้เป็นแคนวาสปริยาย หรือในอีกกรณีหนึ่งหากต้องการให้ผู้ใช้งานมีโอกาสที่จะเห็นข้อมูลในทุกแคนวาสควรตั้งค่าให้มีการวนแคนวาสปริยายไปยังทุก ๆ แคนวาส

4.2.4 กรณีศึกษา 3 : เปรียบเทียบพฤติกรรมของผู้ใช้งานในแต่ละแคนวาส (เมื่อไม่ตั้งแคนวาสปริยาย)

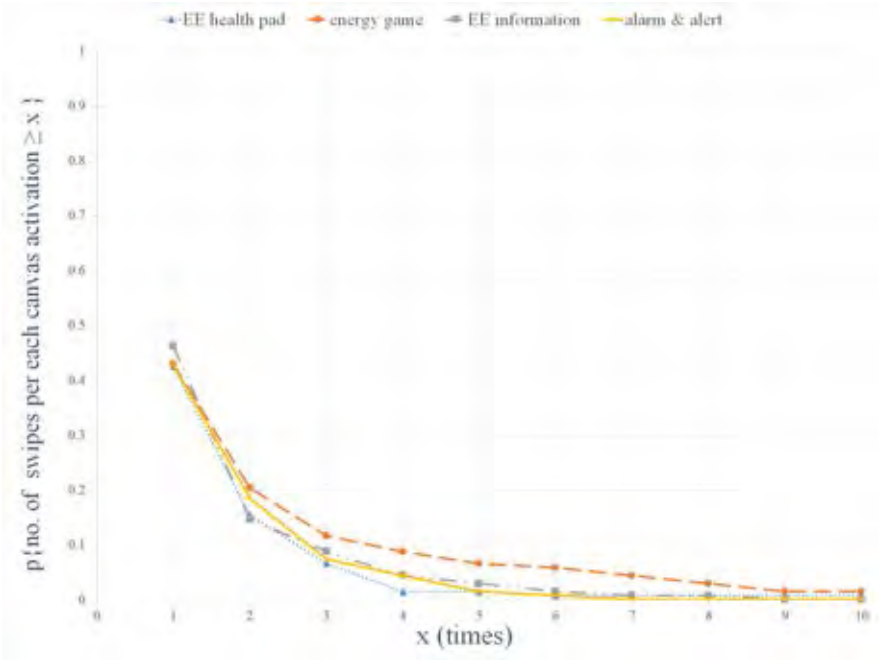
ผู้วิจัยเก็บข้อมูลเพิ่มเติมเป็นเวลา 1 วันโดยตั้งค่าแคนวาสปริยายให้วนไปยังทุก ๆ หมายเลขแคนวาส จากนั้นนำข้อมูลหมายเลขของแคนวาสที่ถูกเรียกใช้งานซึ่งมีจำนวนการเข้าใช้งานดังแสดงในตารางที่ 4.9 และข้อมูลจำนวนการโบกมือของผู้ใช้งานในการเข้าใช้แคนวาสครั้งก่อนหน้ามาวิเคราะห์เพื่อหาว่าผู้ใช้งานมีการโต้ตอบกับโปรแกรมในแคนวาสใดมากที่สุดโดยหาความถี่ของการเรียกใช้งานในหมายเลขแคนวาสแต่ละหมายเลขแล้วโบกมือ 1 ครั้ง, 2 ครั้ง ไปจนถึงจำนวนครั้งที่มากที่สุดจากนั้นนำมาเทียบกับจำนวนครั้งที่มีการเรียกใช้งานในหมายเลขแคนวาสแต่ละหมายเลข

ตารางที่ 4.9: จำนวนครั้งที่แคนวาสถูกเรียกใช้งานเมื่อตั้งให้มีการวนแคนวาสปริยาย

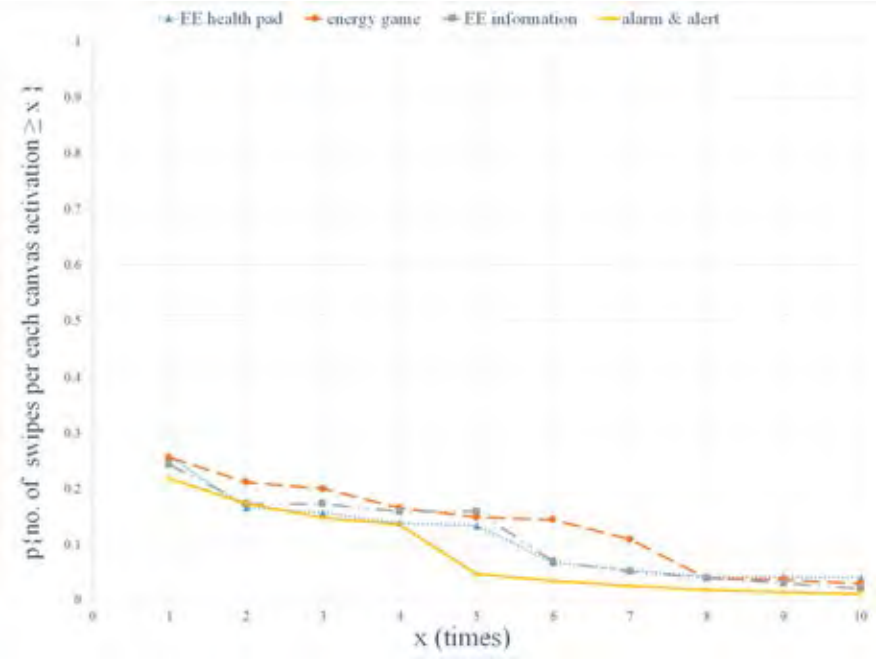
	ชั้น 12 อาคาร 4	ชั้น 13 อาคาร 4	อาคารปฏิบัติการไฟฟ้าแรงสูง
EE health pad	81	136	426
energy game	85	137	429
EE information	80	136	427
alarm & alert	79	135	425



รูปที่ 4.41: ร้อยละของผู้ใช้งานที่โต้ตอบกับโปรแกรมในแต่ละแคนวาสของบริเวณชั้น 12 อาคารเจริญวิศวกรรม



รูปที่ 4.42: ร้อยละของผู้ใช้งานที่โต้ตอบกับโปรแกรมในแต่ละแคนวาสของบริเวณชั้น 13 อาคารเจริญวิศวกรรม



รูปที่ 4.43: ร้อยละของผู้ใช้งานที่โต้ตอบกับโปรแกรมในแต่ละแคนวาสของบริเวณอาคารปฏิบัติการไฟฟ้าแรงสูง

ผลจากกการศึกษาข้อมูลในข้างต้นพบว่าข้อมูลของการโต้ตอบของผู้ใช้งานกับโปรแกรมในแต่ละแคนวาสเป็นไปดังรูปที่ 4.41 ถึง 4.43 ซึ่งเป็นข้อมูลของบริเวณชั้น 12 อาคารเจริญวิศวกรรม (ตึก 4), ชั้น 13 อาคารเจริญวิศวกรรม (ตึก 4) และบริเวณหน้าอาคารปฏิบัติการไฟฟ้าแรงสูงตามลำดับโดยจะเห็นว่าว่าผู้ใช้งานมีการโต้ตอบกับโปรแกรมในแคนวาสเกมส์แสดงการใช้พลังงานมากที่สุดในทุกจุดติดตั้ง ในส่วนของแคนวาสอื่น ๆ ผู้ใช้งานมีการโต้ตอบใกล้เคียงกันหมด

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 บทสรุป

วิทยานิพนธ์ฉบับนี้ได้พัฒนาโปรแกรมสร้างมโนภาพเชิงโต้ตอบด้วยตัวรับรู้ท่าทางสำหรับระบบจัดการพลังงานในอาคารของโครงการ CU-BEMS โดยโปรแกรมสร้างมโนภาพนี้พัฒนาขึ้นจากการเขียนโปรแกรมด้วยภาษา processing ที่ทำงานร่วมกับตัวรับรู้ท่าทาง kinect โดยเมื่อผู้ใช้งานเดินผ่านเข้ามาในบริเวณที่ตัวรับรู้ท่าทางสามารถตรวจจับได้ kinect จะส่งอินพุตจากผู้ใช้ไปยัง SimpleOpenNI ซึ่งเป็นคลัง (โปรแกรม) ในโปรแกรม processing เพื่อตรวจจับผู้ใช้ จากนั้นเมื่อตรวจจับผู้ใช้สำเร็จ SimpleOpenNI จะติดตามโครงกระดูกของผู้ใช้งานเพื่อให้โปรแกรม processing สามารถเรียกใช้ข้อมูลข้อต่อของโครงกระดกเพื่อให้ผู้ใช้งานสามารถใช้ท่าทางในการควบคุมโปรแกรมสร้างมโนภาพได้

วิทยานิพนธ์ฉบับนี้ได้ประยุกต์ใช้มาตรฐาน IEEE1888 ในการสื่อสารระหว่างโปรแกรมที่พัฒนาขึ้นกับหน่วยเก็บข้อมูลของโครงการโดยใช้โพรโทคอล FETCH ในการร้องขอข้อมูลการใช้พลังงานไฟฟ้าเพื่อนำเสนอในส่วนแสดงผลหลักของโปรแกรม และใช้โพรโทคอล WRITE ในการเขียนข้อมูลพฤติกรรมการใช้งานของผู้ใช้งานได้แก่จำนวนการโบกมือของผู้ใช้งานในการเข้าใช้แคนวาสครั้งก่อนหน้า, หมายเลขของแคนวาสที่ถูกเรียกใช้งาน และจำนวนครั้งที่ระบบถูกกระตุ้นให้เข้าสู่แคนวาสปริยายจากคนที่เดินผ่าน โดยข้อมูลเหล่านี้จะถูกเขียนไปยังหน่วยเก็บข้อมูลของโครงการ CU-BEMS เพื่อใช้ในการประเมินพฤติกรรมของผู้ใช้งานที่มีต่อระบบที่พัฒนาขึ้น

การติดตั้งเพื่อการใช้งานจริงจำเป็นต้องมีการพัฒนาโปรแกรมต้นแบบเพื่อทดสอบการทำงานพื้นฐานรวมทั้งประเมินความพึงพอใจของผู้ใช้งานเบื้องต้น และจากการทดสอบเบื้องต้นในบทที่ 3 พบว่าการทำงานร่วมกันระหว่างส่วนประกอบต่าง ๆ ภายในระบบเป็นไปตามหน้าที่ ในส่วนของการประเมินความพึงพอใจหลังการใช้งานโปรแกรมต้นแบบได้นำวิธีการประเมินด้วย Likert scale มาใช้ โดยการให้คะแนนเริ่มจาก 1 คือเห็นด้วยอย่างยิ่ง ไปจนถึง 5 คือไม่เห็นด้วยอย่างยิ่ง ซึ่งผลจากการประเมินนั้นค่าเฉลี่ยของความพึงพอใจจากผู้ใช้งานเป็นที่ยอมรับได้กล่าวคือคะแนนอยู่ในช่วงระหว่าง 1 ไปจนถึง 3 คะแนน การประเมินพฤติกรรมของผู้ใช้งานจากการติดตั้งเพื่อใช้งานจริงซึ่งอุปกรณ์ของระบบที่พัฒนาขึ้นถูกติดตั้งไว้บริเวณภาควิชาวิศวกรรมไฟฟ้ารวมทั้งสิ้น 3 จุด คือบริเวณโถงลิฟท์ชั้น 12 และชั้น 13 ของอาคารเจริญวิศวกรรม รวมทั้งบริเวณหน้าทางเข้าอาคารปฏิบัติการไฟฟ้าแรงสูง ผลจากเก็บข้อมูลเป็นดังนี้

1. การเปรียบเทียบพฤติกรรมผู้ใช้งานในแต่ละพื้นที่ติดตั้งใน 1 สัปดาห์พบว่าบริเวณหน้าอาคารปฏิบัติการไฟฟ้าแรงสูงจุดที่ระบบถูกกระตุ้นให้เข้าสู่แคนวาสปริยายจากคนที่เดินผ่านมากที่สุดซึ่งมีจำนวน 4950 ครั้ง ตามด้วยบริเวณชั้น 13 อาคารเจริญวิศกรรมซึ่งมีจำนวน 1561 ครั้ง และบริเวณชั้น 12 อาคารเจริญวิศกรรมซึ่งมีจำนวน 1301 ครั้ง โดยจากข้อมูลนี้เมื่อนำมาใช้ร่วมกับข้อมูลจำนวนการโบกมือของผู้ใช้งานในการเข้าใช้แคนวาสครั้งก่อนหน้าเพื่อหาว่าหลังจากระบบถูกกระตุ้นจากคนที่เดินผ่านนั้นได้รับความสนใจหรือไม่จะพบว่าเมื่อเริ่มนับจากคนที่เดินผ่านมาแล้วโบกมือ 1 ครั้งขึ้นไปบริเวณชั้น 12 อาคารเจริญวิศวกรรมได้รับความสนใจสุงสุดโดยคิดเป็นร้อยละ 54.2 บริเวณชั้น 13 อาคารเจริญวิศวกรรมระบบได้รับความสนใจคิดเป็น

จำนวนร้อยละ 44.6 และบริเวณอาคารปฏิบัติการไฟฟ้าแรงสูงระบบได้รับความสนใจคิดเป็นจำนวนร้อยละ 37.8

2. การเปรียบเทียบพฤติกรรมของผู้ใช้งานในแต่ละแคนวาส (เมื่อมีการตั้งแคนวาสปริยาย) พบว่าเมื่อนำข้อมูลหมายเลขแคนวาสที่มีการเรียกใช้งาน ใน 1 สัปดาห์มาทำการศึกษาพฤติกรรมการของผู้ใช้งานพบว่าแคนวาสแจ้งเตือนการใช้พลังงานไฟฟ้ามีการเข้าใช้งานมากที่สุดในทุก ๆ พื้นที่ติดตั้งเนื่องจากถูกตั้งค่าไว้เป็นแคนวาสปริยายของระบบดังนั้นทุกครั้งที่คนเดินผ่านจะถูกตั้งค่าให้เปลี่ยนจากแคนวาสสุดซึ่งทำหน้าที่เป็น screen saver มายังแคนวาสซึ่งถูกตั้งค่าไว้ให้ค่าปริยายโดยอัตโนมัติทำให้ผู้ใช้งานมีโอกาสที่จะเห็นข้อมูลในแคนวาสปริยายมากที่สุด ผู้วิจัยจึงขอแสดงความเห็นว่าหากต้องการนำเสนอข้อมูลใดเป็นข้อมูลหลักให้กับผู้ใช้งานควรตั้งค่าไว้เป็นแคนวาสปริยาย หรือในอีกกรณีหนึ่งหากต้องการให้ผู้ใช้งานมีโอกาสที่จะเห็นข้อมูลในทุกแคนวาสควรตั้งค่าให้มีการวนแคนวาสปริยายไปยังทุก ๆ แคนวาส

3. การเปรียบเทียบพฤติกรรมของผู้ใช้งานในแต่ละแคนวาส (เมื่อไม่ตั้งแคนวาสปริยาย) เมื่อเก็บข้อมูลเพิ่มเติมเป็นเวลา 1 วันโดยตั้งค่าแคนวาสปริยายให้วนไปยังทุก ๆ หมายเลขแคนวาสจากนั้นนำข้อมูลหมายเลขของแคนวาสที่ถูกเรียกใช้งานซึ่งมีจำนวนการเรียกใช้งานร่วมกับข้อมูลจำนวนการโบกมือของผู้ใช้งานในการเข้าใช้แคนวาสครั้งก่อนหน้ามาวิเคราะห์เพื่อหาว่าผู้ใช้งานมีการโต้ตอบกับโปรแกรมในแคนวาสใดมากที่สุดพบว่าผู้ใช้งาน มีการโต้ตอบกับโปรแกรมในแคนวาสของเกมส์แสดงการใช้พลังงานมากที่สุดในทุกจุดติดตั้ง ในส่วนของแคนวาสอื่น ๆ ผู้ใช้งานมีการโต้ตอบใกล้เคียงกันหมด

5.2 ข้อเสนอแนะ

การพัฒนาเพื่อเพิ่มประสิทธิภาพการทำงานของระบบให้ดีขึ้นในอนาคต ได้แก่

1. การปรับปรุงเสถียรภาพของการสื่อสารกันรหว่าง front-end และ back-end โดยพัฒนาโปรแกรมให้ยังสามารถทำงานได้ตามหน้าที่ในขณะที่โครงข่ายอินเทอร์เน็ตหรือหน่วยเก็บข้อมูลของโครงการ CU-BEMS เกิดปัญหาขัดข้อง

2. การพัฒนารูปแบบการนำเสนอข้อมูลเพื่อเพิ่มความน่าสนใจให้กับระบบที่พัฒนาขึ้นโดยใช้สื่อประสม เช่น การใช้วีดีทัศน์ และเสียงมาใช้ร่วมกับการนำเสนอข้อมูล

3. การเพิ่มเติมชนิดของข้อมูลที่ใช้นำเสนอนอกจากเหนือข้อมูลการใช้พลังงานไฟฟ้า เช่น ข้อมูลอุณหภูมิ ความชื้นสัมพัทธ์ ความเข้มแสง

4. การปรับปรุงเสถียรภาพของการตรวจจับ และการโต้ตอบระหว่างผู้ใช้งานกับระบบที่พัฒนาขึ้นโดยพัฒนาโปรแกรมให้สามารถใช้งานกลางแจ้งได้รวมถึงพัฒนารูปแบบการโต้ตอบระหว่างผู้ใช้งานและระบบที่พัฒนาขึ้นเพิ่มเติมจากการโบกมือเพียงอย่างเดียว

รายการอ้างอิง

[1] กระทรวงพลังงาน. แผนอนุรักษ์พลังงาน 20 ปี, พฤษภาคม 2554.

[2] ISO. Building Automation and Control Systems. IS0 16484-5, 2003.

[3] Alonso, J. M., Ribas, J., Del Coz, J. J., Calleja, A. J., Lopez, E., and Rico-Secades, M. Intelligent Control System for Fluorescent Lighting Based on LonWorks Technology. in Proceeding of IEEE Industrial Electronics Society, 1998.

[4] Song, X. H., Lu, M., Wu, M., Wang, H., and Lui, F. The Solution of Hybrid Electric Vehicle Information System by Modbus Protocol. in Proceeding of International Conference on Electric Information and Control Engineering, 2011.

[5] IEEE. Standard for Ubiquitous Green Community Control Network. IEEE1888-2011, 2011.

[6] Ninagawa, C., Yoshida, H., Kondo, S., and Otake, H. Data Transmission of IEEE1888 Communication for Wide-area Real-time Smart Grid Application. in Proceeding of IEEE Renewable and Sustainable Energy Conference, 2013.

[7] Esaki, H., and Ochiai, H. GUTP and IEEE1888 for Smart Facility System using Internet Architecture Framework. in 1st IEEE Workshop on Holistic Building Intelligence through Sensing Systems, 2011.

[8] ธนากร อินทสุทธิ์ และ เชาวน์ดิศ อัศวกุล. การทดสอบระบบการเฝ้าสังเกตสภาพอากาศตามมาตรฐาน IEEE1888 สำหรับระบบการจัดการพลังงานของอาคาร. ใน งานประชุมวิชาการงานวิจัย และพัฒนาเชิงประยุกต์ ครั้งที่ 5, 2013.

[9] Inthasut, T., and Aswakul, C. ZigBee Wireless Sensor Network with IEEE1888 Gateway for Building Energy Management System. in Proceeding of International Conference on Electric Information and Control Engineering, 2014.

[10] Kim, K. S., Kim, H., Heo, T. W., Doh, Y., and Jun, J. A. A Smart Grid Testbed Using Wireless Sensor Networks in a Building. in Proceeding of IEEE International Conference on Sensor Technologies and Applications, 2011.

[11] Kim, K. S., Kim, H., Heo, T. W., and Jun, J. A. Monitoring and Management of Power Consumption in Apartment Using ZigBee. in Proceeding of IEEE International Conference on Sensor Technologies and Applications, 2012.

[12] Becker, B., Kellerer, A., and Schmeck, H. User Interaction Interface for Energy Management in Smart Homes. in Proceeding of IEEE Innovative Smart Grid Technologies, 2012.

[13] Fry, B. Visualizing Data. O'Reilly Media, 2008.

[14] Greenberg, I., Xu, D., and Kumar, D. Processing : Creative Coding and Generative Art in Processing 2. Apress,2013.

[15] D3 [Online]. Avialable from : https://github.com/mbostock/d3/wiki [2014, March]

[16] R [Online]. Avialable from : http://www.r-project.org [2014, March]

[17] Processing [Online]. Avialable from : http://wiki.processing.org [2014, March]

[18] Gephi [Online]. Avialable from : http://en.wikipedia.org/wiki/Gephi [2014, March]

[19] Igoe, T. Making Things Talk. O'Reilly Media, 2007.

[20] Oh, J. H., Kim, S. Y., Kim, S. H., and Shi, C. K. A Study of Interactive Art in the Narrative Form of "Magic Monkey". in Proceedings of the Eight International Conference Computer Graphics Imaging and Visualization, 2011.

[21] Bot, K. D., and Schrauf, R. W. Language Development over the Lifespan. UK, 2009.

[22] Bhuiyan, M., and Picking, R. A Gesture Controlled User Interface for Inclusive Design and Evaluative Study of Its Usability. Journal of Software Engineering and Applications, pp. 513-521, September, 2011.

ภาคผนวก

ก โปรแกรมหลัก (The_Beginning_29092014.pde)

```
import ddf.minim.*;
import SimpleOpenNI.*;
import java.net.*;
import java.io.*;
import java.util.*;

Minim minim_canvas1, minim_canvas2;
AudioPlayer gameBackground, swipeSound, welcome;

////////////////////////////////////////
// initialization of kinect and clock

SimpleOpenNI kinect;
int sec, min, hour, day, weekNumber, month, year;
PFont clock_font, calendar_font, fetch_font, fetch_value_font;
PVector rightHand = new PVector();
PVector convertedrightHand = new PVector();
float preecha_previousMouseX, preecha_previousMouseY, trigger_time;
float hand_x_position, hand_y_position, xMapped, yMapped;
int   [] userList;
float [] userLastTimeSeen = {0, 0, 0, 0, 0, 0} ;
int controlUserId = 1;
// array for collecting the path of picture folder
String[] outList_of_foundImageFiles ;
// set the heightmagin 150 pixel
// for the menu selection
int menuHeight_margin = 150;
int userCount     = 0;    // start counting at 0 user
int canvas_number = 0;    // STart at CanvasNumber 0
int canvas_number_prev;
PImage canvas_Selection1, canvas_Selection1over, canvas_Selection2, canvas_Selection2over,
PImage canvas_Selection3, canvas_Selection3over, canvas_Selection4, canvas_Selection4over;
PImage preecha_bgCanvas1, background_Screensaver, canvas2_background, hand_black, hand_white;

////////////////////////////////
// initialization of canvasmenu

int alreadyRunInitialisationCodeInCurrentCanvas;
// set time at 2 second to select each menu
int preecha_timeovercanvas_selected = 2000;
boolean overPicture1, overPicture2, overPicture3, overPicture4, overClock = false;
int preecha_resetSelect_icon;

//////////////////////////////////////////
// initialization of canvas0 (screensaver)

int preecha_resetIntervakcheck_canvas0;
ArrayList<PImage>preecha_allImages_canvas0;
// path folder to show the picture in this canvas
String preecha_pictureFolder_canvas0 = "/Users/CU-EE/DropBox/CU_EE_fl12/Image/";

//////////////////////////////////////////
// initialization of canvas1 (EE health pad)

ArrayList<PImage> preecha_allImages;
```

```
int    preecha_showPic = 0; // Basis Picture
int    preecha_showPicLeft, preecha_showPicRight;
int    currentImage = 0;
float  preecha_imageShift_xpos = 0;
float  preecha_imageShift_ypos = 0;
float  preecha_imageShift_Target = 0;
// speed to change to next Picture
float  preecha_easingRatio = 0.35;
// path folder to show the picture in this canvas
String preecha_pictureFolder_canvas1 = "/Users/CU-EE/Desktop/"+
"The_Beginning_29092014/healthpad_eeinformation/";

//////////////////////////////////////////
// initialization of canvas2 (energy game)

String fetchGame_res = "";
String [] fetchGame_value;
float  [] ValueGame;
String [] PointGame;
int    [] located;

// point id to retrieving data from fetch
String [] pointID_Game = {
"<key id=\"bems.ee.eng.chula.ac.th/energy_consumption/"+
"department\" attrName=\"time\" select=\"maximum\" />",
"<key id=\"bems.ee.eng.chula.ac.th/energy_consumption/"+
"ee/f2\" attrName=\"time\" select=\"maximum\" />",
"<key id=\"bems.ee.eng.chula.ac.th/energy_consumption/"+
"ee/f3\" attrName=\"time\" select=\"maximum\" />",
"<key id=\"bems.ee.eng.chula.ac.th/energy_consumption/"+
"ee/f4\" attrName=\"time\" select=\"maximum\" />",
"<key id=\"bems.ee.eng.chula.ac.th/energy_consumption/"+
"ee/f5\" attrName=\"time\" select=\"maximum\" />",
"<key id=\"bems.ee.eng.chula.ac.th/energy_consumption/"+
"eng4/f12\" attrName=\"time\" select=\"maximum\" />",
"<key id=\"bems.ee.eng.chula.ac.th/energy_consumption/"+
"eng4/f13\" attrName=\"time\" select=\"maximum\" />",
"<key id=\"bems.ee.eng.chula.ac.th/energy_consumption/"+
"highvoltage\" attrName=\"time\" select=\"maximum\" />"
};

String [] Building_name = {
"EE department",
"EE Building fl2",
"EE Building fl3",
"EE Building fl4",
"EE Building fl5",
"Building4 fl12",
"Building4 fl13",
"High Voltage"
};

// load icon to use in the game
PImage [] icon_Image = {
loadImage("/Users/CU-EE/Desktop/The_Beginning_29092014/gameFolder/eedepartment.jpg"),
loadImage("/Users/CU-EE/Desktop/The_Beginning_29092014/gameFolder/eebulidingfl2.jpg"),
loadImage("/Users/CU-EE/Desktop/The_Beginning_29092014/gameFolder/eebulidingfl3.jpg"),
loadImage("/Users/CU-EE/Desktop/The_Beginning_29092014/gameFolder/eebulidingfl4.jpg"),
```

```
loadImage("/Users/CU-EE/Desktop/The_Beginning_29092014/gameFolder/eebulidingfl5.jpg"),
loadImage("/Users/CU-EE/Desktop/The_Beginning_29092014/gameFolder/eng4_12.jpg"),
loadImage("/Users/CU-EE/Desktop/The_Beginning_29092014/gameFolder/eng4_13.jpg"),
loadImage("/Users/CU-EE/Desktop/The_Beginning_29092014/gameFolder/HVbuilding.jpg")
};

int cak_numrows = 3;
int cak_basis = 1;
int[][] b = new int[cak_numrows][cak_numrows];
int pad = int(70), bs = int(230), len = pad*(b.length+1)+bs*b.length, score = 0, dead = 1;
String heighest_score [] ;
int savedScore;
int max_t = 1; // keeps the max value of score t achieved so far in already-displayed boards
int cak_rightMarginWidth = abs(len-width);
int icon_size  = 100;
int cak_x_icon =   1100;
int cak_y_icon1   = len ;
int cak_y_icon2   = cak_y_icon1 - (icon_size+10);
int cak_y_icon4   = cak_y_icon2 - (icon_size+10);
int cak_y_icon8   = cak_y_icon4 - (icon_size+10);
int cak_y_icon16  = cak_y_icon8 - (icon_size+10);
int cak_y_icon32  = cak_y_icon16 - (icon_size+10);
int cak_y_icon64  = cak_y_icon32 - (icon_size+10);
int cak_y_icon128 = cak_y_icon64 - (icon_size+10);
PImage icon1_image, icon2_image, icon4_image, icon8_image;
PImage icon16_image, icon32_image, icon64_image, icon128_image;

//////////////////////////////////////////
// initialization of canvas3 (EE information)

ArrayList<PImage> preecha_allImages_canvas3 ;
int    preecha_showPic_canvas3 = 0;
int    preecha_showPicLeft_canvas3, preecha_showPicRight_canvas3;
int    currentImage_canvas3 = 0;
float  preecha_imageShift_xpos_canvas3 = 0;
float  preecha_imageShift_ypos_canvas3 = 0;
float  preecha_imageShift_Target_canvas3 = 0;
// speed to change to next Picture
float  preecha_easingRatio_canvas3 = 0.35;
// path folder to show the picture in this canvas
String preecha_pictureFolder_canvas3 ="/Users/CU-EE/Dropbox/CU_EE_fl12/EE_information/";

//////////////////////////////////////////
// initialization of canvas3 (EE information)

PFont alarm, textwarn, textBuilding;
String [] Building_name_alarm = {
"\"Electical Engineering Department\"",
"\"EE Building floor 2\"",
"\"EE Building floor 3\"",
"\"EE Building floor 4\"",
"\"EE Building floor 5\"",
"\"Building 4 floor 12\"",
"\"Building 4 floor 13\"",
"\"High Voltage Building\""
};
// index to show in each installation area
// from the Building_name_alarm array
```

```
// 5 is mean this is for ENG4 fl12
int zone_index = 5;
int percent;
int [] target;
int saturday, sunday;
float radiusOfEnergyConsumed;
float radius, fontBD, percentFontRatio;
//path of the  quota target file
String [] tarketQuata =loadStrings("/Users/CU-EE/Dropbox/tarketQouta.csv");

////////////////////////////
// initialization of backend

String [] fetch_value;
ArrayList<String>  PointID;
ArrayList<Float> Value;
String [] timeStamp_cv = {};
String [] canvas_num = {};
String [] cummSwipe = {};
int noSwipe, noUser = 0;
String write_sec, write_min, write_hour, write_day, write_month, write_year;
String [] pointID = {
"<key id=\"bems.ee.eng.chula.ac.th/ee_health_pad/"+
"co2_reduction\" attrName=\"time\" select=\"maximum\" />",
"<key id=\"bems.ee.eng.chula.ac.th/ee_health_pad/"+
"coal_reduction\" attrName=\"time\" select=\"maximum\" />",
"<key id=\"bems.ee.eng.chula.ac.th/ee_health_pad/"+
"consumption\" attrName=\"time\" select=\"maximum\" />",
"<key id=\"bems.ee.eng.chula.ac.th/ee_health_pad/"+
"electricity_saved\" attrName=\"time\" select=\"maximum\" />",
"<key id=\"bems.ee.eng.chula.ac.th/ee_health_pad/"+
"energy_saved\" attrName=\"time\" select=\"maximum\" />",
"<key id=\"bems.ee.eng.chula.ac.th/ee_health_pad/"+
"load_factor\" attrName=\"time\" select=\"maximum\" />",
"<key id=\"bems.ee.eng.chula.ac.th/ee_health_pad/"+
"peak_saved\" attrName=\"time\" select=\"maximum\" />",
"<key id=\"bems.ee.eng.chula.ac.th/ee_health_pad/"+
"renewable_energy\" attrName=\"time\" select=\"maximum\" />",
"<key id=\"bems.ee.eng.chula.ac.th/ee_health_pad/"+
"tree_saved\" attrName=\"time\" select=\"maximum\" />"
};
String[] target_healthpad = {
"10", "594.4", "0.8", "10,000", "1000", "99", "50", "n/a", "50"
};
String[] target_healthpad_unit = {
"Tons CO2", "kg", "kWh/m^2", "baht", "kWh", "%", "kW", "", "trees"
};


////////////////////////////
// start settingup program

void setup() {
  size(displayWidth, displayHeight);

  // user Trackig SET UP //
  kinect = new SimpleOpenNI(this);
  kinect.enableDepth();
```

```
  kinect.enableRGB();
  kinect.enableUser();

  // for loading music background in canvas 2
  minim_canvas2   = new Minim(this);
  gameBackground = minim_canvas2.loadFile("bgm.mp3", 2048);

  ////////////////
  // loading font
  clock_font       = loadFont("CordiaUPC-Bold-200.vlw");
  fetch_font       = loadFont("Calibri-Light-200.vlw");
  fetch_value_font = loadFont("Calibri-Bold-200.vlw");
  calendar_font    = loadFont("Microsoft-Yi-Baiti-48.vlw");
  alarm            = loadFont("ArialRoundedMTBold-200.vlw");
  textwarn         = loadFont("CordiaUPC-100.vlw");
  textBuilding     = loadFont("CordiaUPC-Bold-150.vlw");

  ////////////////////////////
  // loading image of menu icon
  canvas_Selection1     = loadImage("/center/HealthPad.png");
  canvas_Selection1over = loadImage("/center/HealthPad_click.png");
  canvas_Selection2     = loadImage("/center/EnergyGame1.png");
  canvas_Selection2over = loadImage("/center/EnergyGame.png");
  canvas_Selection3     = loadImage("/center/EEinf.png");
  canvas_Selection3over = loadImage("/center/EEinf_click.png");
  canvas_Selection4     = loadImage("/center/Alarm.png");
  canvas_Selection4over = loadImage("/center/Alarm_click.png");

  ////////////////////////////
  // loading image of hand icon
  hand_black = loadImage("/center/hand_black.png");
  hand_black.resize(100, 100);
  hand_white  = loadImage("/center/hand_white.png");
  hand_white.resize(100, 100);

  canvas2_background = loadImage("/Users/CU-EE/Desktop/"+
  "The_Beginning_29092014/canvas2/canvas2_background.jpg");
  canvas2_background.resize(width, height-menuHeight_margin);
  ieee1888_fetch();
  ieee1888_fetch_Game();
  background_Screensaver =loadImage("background_canvas0.jpg");
  image(background_Screensaver, 0, 0);
  noCursor();
}


////////////////////
// start draw loop
//

void draw() {

  ////////////////////////////////////////////////////
  // keep tracking and check user status from kinect
  //

  trackingUser();
  hand_x_position = detect_hand_x_position();
```

```
hand_y_position = detect_hand_y_position();

////////////////////////////////////////////
// check if canvas number has been changed
//

if (canvas_number_prev != canvas_number) {

  // write canvas_number_prev, canvas_number, noSwipe, userList.length, time
  ieee1888_write();

  // reset all per-canvas counters //
  noSwipe = 0;
  alreadyRunInitialisationCodeInCurrentCanvas = 0;

  // stop playing background sound of previous canvas //
  gameBackground.close();
  gameBackground = minim_canvas2.loadFile("bgm.mp3", 2048);
}

canvas_number_prev = canvas_number;

// no users detected so reset our screen to canvas number 0 and reset swipe counter //
if (userCount == 0) {
  canvas_number = 0;
  zone_index = 5;
}

// fresh user has just entered so switch from canvas numbers 0 to 1 //
if ( (userCount > 0) && (canvas_number_prev == 0) ) {
  canvas_number = 4;
  noUser +=1;

  // fetch data from data storage to use in each canvas
  ieee1888_fetch();
  ieee1888_fetch_Game();
}

if (canvas_number == 0) {
  canVas0();
}
if (canvas_number == 1) {
  canVas_Menu();
  canVas1();
  checkMouseCanVas1(int(hand_x_position), int(hand_y_position));
}
if (canvas_number == 2) {
  canVas_Menu();
  canVas2();
  checkMouseCanVas2(int(hand_x_position), int(hand_y_position));
}
if (canvas_number == 3) {
  canVas_Menu();
  canVas3();
  checkMouseCanVas3(int(hand_x_position), int(hand_y_position));
}
if (canvas_number == 4) {
  canVas_Menu();
```

```
    canVas4();
    checkMouseCanVas4(int(hand_x_position), int(hand_y_position));
  }
}
```

ข โปรแกรมเพื่อใช้ในการติดต่อระหว่าง front-end กับ back-end (backend.pde)

```
////////////////////////////////////
// IEEE1888 FETCH for EE health pad
//
void ieee1888_fetch() {
  try {
    String xmldata = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>" +
    "<soapenv:Envelope " +
    "xmlns:soapenv="+
    "\"http://schemas.xmlsoap.org/soap/envelope/\">"+
    "<soapenv:Body>"+
    "<ns2:queryRQ xmlns:ns2=\"http://soap.fiap.org/\">"+
    "<transport xmlns=\"http://gutp.jp/fiap/2009/11/\">"+
    "<header>"+
    "<query id=\"9eed9de4-1c48-4b08-a41d-dac067fc1c0d\" type=\"storage\">"+
    pointID[0]+pointID[1]+pointID[2]+pointID[3]+pointID[4]+
    pointID[5]+pointID[6]+pointID[7]+pointID[8]+
    "</query></header></transport></ns2:queryRQ>"+
    "</soapenv:Body></soapenv:Envelope>";

    //Create socket
    String hostname = "161.200.90.122";
    int port = 80;
    InetAddress  addr = InetAddress.getByName(hostname);
    Socket sock = new Socket(addr, port);

    //Send header
    String path = "/axis2/services/FIAPStorage";
    BufferedWriter  wr =
    new BufferedWriter(new OutputStreamWriter(sock.getOutputStream(),"UTF-8"));
    wr.write("POST " + path + " HTTP/1.0\r\n");
    wr.write("Host: 161.200.90.122\r\n");
    wr.write("Content-Length: " + xmldata.length() + "\r\n");
    wr.write("Content-Type: text/xml; charset=\"utf-8\"\r\n");
    wr.write("SOAPAction: http://soap.fiap.org/query\r\n");
    wr.write("\r\n");
    //Send data
    wr.write(xmldata);
    wr.flush();

    // Response
   BufferedReader rd = new BufferedReader(new InputStreamReader(sock.getInputStream()));
    String line;
    while ( (line = rd.readLine ()) != null) {
      fetch_res = fetch_res+line;
    }
  }
  catch (Exception e) {
    e.printStackTrace();
```

```
  }

  ////////////////////////////////
  // start parsing the reponse data
  String [] fetch_body = split(fetch_res, "<body>");
  String [] fetch_pointID = split(fetch_body[1], "</value></point>");
  Value = new ArrayList<Float>();
  PointID = new ArrayList<String>();
  for (int i = 0; i < fetch_pointID.length-1; i++) {
    String [] fetch_value = split(fetch_pointID[i], "+07:00\">");
    PointID.add(fetch_value[0]);
    Value.add(float(fetch_value[1]));
  }
}

////////////////////////////////////////////////
// IEEE1888 FETCH for energygame and alarm alert
//
void ieee1888_fetch_Game() {
  try {
    String xmldata = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>" +
    "<soapenv:Envelope " +
    "xmlns:soapenv="+
    "\"http://schemas.xmlsoap.org/soap/envelope/\">"+
    "<soapenv:Body>"+
    "<ns2:queryRQ xmlns:ns2=\"http://soap.fiap.org/\">"+
    "<transport xmlns=\"http://gutp.jp/fiap/2009/11/\">"+
    "<header>"+
    "<query id=\"9eed9de4-1c48-4b08-a41d-dac067fc1c0d\" type=\"storage\">"+
    pointID_Game[0]+pointID_Game[1]+pointID_Game[2]+
    pointID_Game[3]+pointID_Game[4]+pointID_Game[5]+
    pointID_Game[6]+pointID_Game[7]+
    "</query></header></transport></ns2:queryRQ>"+
    "</soapenv:Body></soapenv:Envelope>";

    //Create socket
    String hostname = "161.200.90.122";
    int port = 80;
    InetAddress  addr = InetAddress.getByName(hostname);
    Socket sock = new Socket(addr, port);

    //Send header
    String path = "/axis2/services/FIAPStorage";
    BufferedWriter  wr
    = new BufferedWriter(new OutputStreamWriter(sock.getOutputStream(), "UTF-8"));
    wr.write("POST " + path + " HTTP/1.0\r\n");
    wr.write("Host: 161.200.90.122\r\n");
    wr.write("Content-Length: " + xmldata.length() + "\r\n");
    wr.write("Content-Type: text/xml; charset=\"utf-8\"\r\n");
    wr.write("SOAPAction: http://soap.fiap.org/query\r\n");
    wr.write("\r\n");

    //Send data
    wr.write(xmldata);
    wr.flush();

    // Response
    BufferedReader rd
```

```
    = new BufferedReader(new InputStreamReader(sock.getInputStream()));
    String line;
    while ( (line = rd.readLine ()) != null) {
      fetchGame_res = fetchGame_res+line;
    }
  }

  catch (Exception e) {
    e.printStackTrace();
  }

  ////////////////////////////////
  // start parsing the reponse data
  String [] fetch_body = split(fetchGame_res, "<body>");
  String [] fetch_pointID = split(fetch_body[1], "</value></point>");
  ValueGame = new float[pointID_Game.length];
  PointGame = new String[pointID_Game.length];

  for (int i = 0; i < fetch_pointID.length-1; i++) {
    String [] fetchGame_value = split(fetch_pointID[i], "+07:00\">");
    ValueGame[i] = float(fetchGame_value[1]);
    PointGame[i] = fetchGame_value[0];
  }
  float[] copyValueGame = new float[ValueGame.length];
  arrayCopy(ValueGame, copyValueGame);
  located = new int[copyValueGame.length];
  for (int i = 0; i < copyValueGame.length; i++) {
    float check_Max = max(copyValueGame);
    int minIndex = 0;
    for (int j = 0; j < copyValueGame.length; j++) {
      if (check_Max > copyValueGame[j]) {
        minIndex = j;
        check_Max = copyValueGame[j];
      }
    }
    copyValueGame[minIndex] = max(copyValueGame);
    located[i] = minIndex;
  }
}

//////////////////
// IEEE1888 WRITE
//
void ieee1888_write() {
  write_year  = ""+year();   // 2003, 2004, 2005, etc.
  if (month() < 10) {
    write_month = "0"+month();
  } else {
    write_month = ""+month();
  }
  if (day() < 10) {
    write_day = "0"+day();
  } else {
    write_day = ""+day();
  }
  if (hour() < 10) {
    write_hour = "0"+hour();
  } else {
```

```
  write_hour = ""+hour();
}
if (minute() < 10) {
  write_min = "0"+minute();
} else {
  write_min = ""+minute();
}
if (second() < 10) {
  write_sec = "0"+second();
} else {
  write_sec = ""+second();
}

try {
  String xmldata = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>" +
  "<soapenv:Envelope " +
  "xmlns:soapenv="+
  "\"http://schemas.xmlsoap.org/soap/envelope/\">"+
  "<soapenv:Body>"+
  "<ns2:dataRQ xmlns:ns2=\"http://soap.fiap.org/\">"+
  "<transport xmlns=\"http://gutp.jp/fiap/2009/11/\">"+
  "<body>"+
  "<pointSet id=\"http://bems.ee.eng.chula.ac.th/eng4/"+
  "fl12/corridor/elevatorfront/kinect\">"+
  "<point id=\"http://bems.ee.eng.chula.ac.th/eng4/"+
  "fl12/corridor/elevatorfront/kinect/num_user\">"+
  "<value time=\""+write_year+"-"+write_month+"-"+write_day+"T"+
  write_hour+":"+write_min+":"+write_sec+".000+07:00\">"+
  noUser+"</value>"+"</point>"+
  "<point id=\"http://bems.ee.eng.chula.ac.th/eng4/"+
  "fl12/corridor/elevatorfront/kinect/canvas_num\">"+
  "<value time=\""+write_year+"-"+write_month+"-"+write_day+"T"+
  write_hour+":"+write_min+":"+write_sec+".000+07:00\">"+
  "canvas"+canvas_number+"</value>"+"</point>"+
  "<point id=\"http://bems.ee.eng.chula.ac.th/eng4/"+
  "fl12/corridor/elevatorfront/kinect/num_swipe\">"+
  "<value time=\""+write_year+"-"+write_month+"-"+write_day+"T"+
  write_hour+":"+write_min+":"+write_sec+".000+07:00\">"+
  "swipe"+noSwipe+"</value>"+
  "</point>"+"</pointSet>"+"</body>"+
  "</transport></ns2:dataRQ>"+
  "</soapenv:Body></soapenv:Envelope>";

  //Create socket
  String hostname = "161.200.90.122";
  int port = 80;
  InetAddress  addr = InetAddress.getByName(hostname);
  Socket sock = new Socket(addr, port);

  //Send header
  String path = "/axis2/services/FIAPStorage";
  BufferedWriter  wr
  = new BufferedWriter(new OutputStreamWriter(sock.getOutputStream(), "UTF-8"));
  wr.write("POST " + path + " HTTP/1.0\r\n");
  wr.write("Host: 161.200.90.122\r\n");
  wr.write("Content-Length: " + xmldata.length() + "\r\n");
  wr.write("Content-Type: text/xml; charset=\"utf-8\"\r\n");
  wr.write("SOAPAction: http://soap.fiap.org/data\r\n");
```

```
    wr.write("\r\n");

    //Send data
    wr.write(xmldata);
    wr.flush();

    // Response
    BufferedReader rd
    = new BufferedReader(new InputStreamReader(sock.getInputStream()));
    String line;
    while ( (line = rd.readLine ()) != null) {
    }
  }
  catch (Exception e) {
    e.printStackTrace();
  }
}
```

ค โปรแกรมในส่วนแสดงผลของหน้า screen saver (canvas0.pde, canvas0_init.pde)

```
void canVas0() {
  if (alreadyRunInitialisationCodeInCurrentCanvas == 0) {

    // run the following canvas-initialisation code only //
    // once per each time of entering this canvas        //
    canvas0_init();
    alreadyRunInitialisationCodeInCurrentCanvas = 1;
  }
  try {
    int preecha_passTimecanvas0 = millis()-preecha_resetIntervakcheck_canvas0;
    if (preecha_passTimecanvas0 > 3000 ) {
    float imageRotation = random(-50, 50);
    float imageX = random(250, width-250);
    float imageY = random(200, height-200);
    ////////////////////////////
    // Draw, Rotate + Scale Image
    imageMode(CENTER);
    pushMatrix();
    translate(imageX, imageY);
    rotate(radians(imageRotation));
    filter(BLUR, .8);
    filter(GRAY);
    filter(INVERT);
    currentImage +=1;
    image(preecha_allImages_canvas0.get
    (currentImage%preecha_allImages_canvas0.size()),0,0);
    popMatrix();
    preecha_resetIntervakcheck_canvas0 = millis();
    }
  }
  catch (Exception e){
    preecha_allImages_canvas0 = new ArrayList<PImage>();
    preecha_checkImgfiles(preecha_pictureFolder_canvas0);
  }
}
```

```
//////////////////
// canvas0 init
void canvas0_init() {
  translate(0, 0);
  imageMode(CORNER);
  image(background_Screensaver, 0, 0);
}
```

ง โปรแกรมในส่วนแสดงผลของแคนวาส EE healthpad (canvas1.pde)

```
void canVas1() {
try{
 pushMatrix();
 translate(preecha_imageShift_xpos, preecha_imageShift_ypos+150);
 if (preecha_showPic == 0) {
 preecha_showPicLeft = (preecha_allImages.size()) - 1;
 } else {
 preecha_showPicLeft = preecha_showPic - 1;
 }

 if (preecha_showPic == (preecha_allImages.size()) - 1 ) {
 preecha_showPicRight = 0;
 } else {
 preecha_showPicRight = preecha_showPic + 1;
 }

 PImage picLeft = preecha_allImages.get(preecha_showPicLeft);
 PImage pic = preecha_allImages.get(preecha_showPic);
 PImage picRight = preecha_allImages.get(preecha_showPicRight);

 image (picLeft, -width, 0);
 image (pic, 0, 0);
 image (picRight, width, 0);
 fill(255);
 textFont(fetch_font, 90);
 textAlign(LEFT);
 text("TARGET", (width/2)+100, 150);
 text("CURRENT", (width/2)+100, 500);
 textFont(fetch_value_font, 120);
 textAlign(LEFT);
 text(target_healthpad[preecha_showPic]+" "+
 target_healthpad_unit[preecha_showPic], (width/2)+100, 300);
 text(String.format("%.1f", Value.get(preecha_showPic))+" "+
 target_healthpad_unit[preecha_showPic],(width/2)+100, 650);
  }
 catch (Exception e) {
  preecha_allImages = new ArrayList<PImage>();
  preecha_checkImgfiles(preecha_pictureFolder_canvas1);
 }
  preecha_imageShift_xpos +=(preecha_imageShift_Target-
  preecha_imageShift_xpos)*preecha_easingRatio;
  popMatrix();
}


//////////////
```

```
// hand event
//
void checkMouseCanVas1(int hand_x_position, int hand_y_position) {
  mouseX = hand_x_position ;
  mouseY = hand_y_position ;
  preecha_previousMouseX = lerp(preecha_previousMouseX, mouseX, 0.09);

  float current_time = millis();
  int dx;
  if (current_time - trigger_time > 2500 ) {
    image(hand_black, mouseX, mouseY);
    // check condition for trigger hand event
    dx=(preecha_previousMouseX-mouseX)>=150?-1:
    ((preecha_previousMouseX-mouseX)<=-150 ? 1 : 0);
    if (dx == -1) {
      noSwipe +=1;
      preecha_showPic = (preecha_showPic+1)%(preecha_allImages.size());
      preecha_imageShift_xpos = width;
    }
    if (dx == 1) {
      noSwipe +=1;
      if (preecha_showPic == 0) {
        preecha_showPic = (preecha_allImages.size())-1;
      }
      else {
        preecha_showPic = (preecha_showPic-1);
      }
      preecha_imageShift_xpos = -width;
    }
    if (dx != 0) {
      trigger_time = millis();
    }
  }
  else {
    image(hand_white, mouseX, mouseY);
    dx = 0;
  }
}
```

จ โปรแกรมในส่วนแสดงผลของแคนวาส energy game (canvas2_init.pde,canvas2.pde)

```
////////////////////////////
// setting up an energy game
//
void canvas2_init() {
  max_t = 1;
  restart();
  gameBackground.play();
}

//////////////////////
// start an energy game
//
void canVas2() {
    if (alreadyRunInitialisationCodeInCurrentCanvas == 0) {
```

```
  // run the following canvas-initialisation code only //
  // once per each time of entering this canvas        //
  alreadyRunInitialisationCodeInCurrentCanvas = 1;
}
noStroke();
String heighest_score [] = loadStrings("score.txt");
int savedScore = int(heighest_score[0]);
cak_icondisp_background(max_t);

for (int j = 0; j < b.length; j++)
  for (int i = 0; i < b[j].length; i++) {
    fill(200);
    rect((pad+(pad+bs)*i)+pad, (pad+(pad+bs)*j)+menuHeight_margin-20, bs, bs, 30);
  }
for (int j = 0; j < b.length; j++)
  for (int i = 0; i < b[j].length; i++) {
    float x = (pad+(pad+bs)*i)+pad, y=(pad+(pad+bs)*j)+menuHeight_margin-20;
    if (b[j][i] > 0) {
      float p = log(b[j][i])/log(2);
      cak_icondisp(b[j][i], x, y, bs);
      max_t = max(max_t, b[j][i]);
    }
  }
noStroke();
fill(51, 102, 204);
rect(0, menuHeight_margin, width, pad-30);
textt("SCORE : "+score, pad, menuHeight_margin+30,
color(255), calendar_font, pad/1.9, LEFT);
textt("kWh (from first day to yesterday of this month)",
width/2, menuHeight_margin+30, color(255), calendar_font, pad/1.9, CENTER);
textt("BEST : "+savedScore, width - pad, menuHeight_margin+30,
color(255), calendar_font, pad/1.9, RIGHT);
noFill();

if (dead>0) {
  fill(200);
  cak_icondisp_gameover(); //cak3: displaying all boards in grey//
  cak_icondisp_background(max_t);

  fill(51, 102, 204);
  rect(pad+60, menuHeight_margin+(2*pad)+bs-30, (3*bs)+(2*pad)+40, bs, 15);
  textt("GAME OVER", ((pad+60)+((3*bs)+(2*pad)+110))/2,
  menuHeight_margin+(3*pad)+bs, color(255), fetch_value_font, 60, CENTER);
  textt("Hold on an \"Energy Game\" icon  3 seconds to restart",
  ((pad+60)+((3*bs)+(2*pad)+110))/2, menuHeight_margin+(3*pad)+bs+70,
  color(255), calendar_font, pad/1.9, CENTER);

  if (overPicture(((0+360)/2)-100, 0, 100, menuHeight_margin)) {
    overPicture1 = true;
    image(canvas_Selection1over, 0, 10);
    image(canvas_Selection2, 360, 6);
    image(canvas_Selection3, 720, 8);
    image(canvas_Selection4, 1080, 8);
  } else if (overPicture(((360+720)/2)-100, 0, 100, menuHeight_margin)) {
    overPicture2 = true;
    image(canvas_Selection1, 0, 10);
    image(canvas_Selection2over, 360, 6);
```

```
      image(canvas_Selection3, 720, 8);
      image(canvas_Selection4, 1080, 8);
    } else if (overPicture(((720+1080)/2)-100, 0, 100, menuHeight_margin)) {
      overPicture3 = true;
      image(canvas_Selection1, 0, 10);
      image(canvas_Selection2, 360, 6);
      image(canvas_Selection3over, 720, 8);
      image(canvas_Selection4, 1080, 8);
    } else if (overPicture(((1080+1440)/2)-100, 0, 100, menuHeight_margin)) {
      overPicture3 = true;
      image(canvas_Selection1, 0, 10);
      image(canvas_Selection2, 360, 6);
      image(canvas_Selection3, 720, 8);
      image(canvas_Selection4over, 1080, 8);
    } else {
      overPicture1 = overPicture2 = overPicture3 = overPicture4 = false;
      image(canvas_Selection1, 0, 10);
      image(canvas_Selection2, 360, 6);
      image(canvas_Selection3, 720, 8);
      image(canvas_Selection4, 1080, 8);
    }

    fill(255, 0, 0);
    ellipse(int(hand_x_position), int(hand_y_position), 20, 20);
    noStroke();
    noFill();

    int newScore = score;
    if (newScore > savedScore ) {

      savedScore = newScore;
      String scoreVal = ""+savedScore;
      String [] scoreFile = {
        scoreVal
      };
      saveStrings("score.txt", scoreFile);
    }
  }
  ellipse(int(hand_x_position), int(hand_y_position), 20, 20);
}

// for show text in the game
void textt(String t, float x, float y, color c, PFont ff, float s, int align) {
  fill(c);
  textAlign(align);
  textFont(ff, s);
  text(t, x, y);
}
// restart game
void restart() {
  b = new int[cak_numrows][cak_numrows];
  spawn();
  score = dead = 0;
}

void spawn() {
  ArrayList<Integer> xs = new ArrayList<Integer>(), ys = new ArrayList<Integer>();
  for (int j = 0; j < b.length; j++) for (int i = 0; i < b[j].length; i++) if (b[j][i]==0) {
```

```
    xs.add(i);
    ys.add(j);
  }
  int rnd = (int)random(0, xs.size()), y = ys.get(rnd), x = xs.get(rnd);
  b[y][x] = random(0, 1) < .9 ? cak_basis : 2*cak_basis;
}

void cak_icondisp(int t, float x, float y, int s) {
  if (t == 1)  {image(icon_Image[located[0]], x, y, s, s);}
  if (t == 2)  {image(icon_Image[located[1]], x, y, s, s);}
  if (t == 4)  {image(icon_Image[located[2]], x, y, s, s);}
  if (t == 8)  {image(icon_Image[located[3]], x, y, s, s);}
  if (t == 16) {image(icon_Image[located[4]], x, y, s, s);}
  if (t == 32) {image(icon_Image[located[5]], x, y, s, s);}
  if (t == 64) {image(icon_Image[located[6]], x, y, s, s);}
  if (t == 128){image(icon_Image[located[7]], x, y, s, s);}
}

void cak_icondisp_background(int max_t) {
  // display icon pyramid on the right margin of game board up to board numbered max_t

  if (max_t >= 1) {
    image(icon_Image[located[0]], cak_x_icon,
    cak_y_icon1, icon_size, icon_size);
    textt(Building_name[located[0]], cak_x_icon+100,
    cak_y_icon1+65, color(160), fetch_font, 40, LEFT);
    textt("kWh", cak_x_icon+550, cak_y_icon1+65,
    color(160), fetch_font, 40, LEFT);
    textt(str(int(ValueGame[located[0]])), cak_x_icon+530,
    cak_y_icon1+65, color(102), fetch_value_font, 60, RIGHT);
  }
  if (max_t >= 2) {
    image(icon_Image[located[1]], cak_x_icon, cak_y_icon2, icon_size, icon_size);
    textt(Building_name[located[1]], cak_x_icon+100,
    cak_y_icon2+65, color(160), fetch_font, 40, LEFT);
    textt("kWh", cak_x_icon+550, cak_y_icon2+65, color(160), fetch_font, 40, LEFT);
    textt(str(int(ValueGame[located[1]])), cak_x_icon+530,
    cak_y_icon2+65, color(102), fetch_value_font, 60, RIGHT);
  }
  if (max_t >= 4) {
    image(icon_Image[located[2]], cak_x_icon, cak_y_icon4, icon_size, icon_size);
    textt(Building_name[located[2]], cak_x_icon+100,
    cak_y_icon4+65, color(160), fetch_font, 40, LEFT);
    textt("kWh", cak_x_icon+550, cak_y_icon4+65, color(160), fetch_font, 40, LEFT);
    textt(str(int(ValueGame[located[2]])), cak_x_icon+530,
    cak_y_icon4+65, color(102), fetch_value_font, 60, RIGHT);
  }
  if (max_t >= 8) {
    image(icon_Image[located[3]], cak_x_icon, cak_y_icon8, icon_size, icon_size);
    textt(Building_name[located[3]], cak_x_icon+100,
    cak_y_icon8+65, color(160), fetch_font, 40, LEFT);
    textt("kWh", cak_x_icon+550, cak_y_icon8+65,color(160), fetch_font, 40, LEFT);
    textt(str(int(ValueGame[located[3]])), cak_x_icon+530,
    cak_y_icon8+65, color(102), fetch_value_font, 60, RIGHT);
  }
  if (max_t >= 16) {
    image(icon_Image[located[4]], cak_x_icon, cak_y_icon16, icon_size, icon_size);
    textt(Building_name[located[4]], cak_x_icon+100,
```

```
    cak_y_icon16+65, color(160), fetch_font, 40, LEFT);
    textt("kWh", cak_x_icon+550, cak_y_icon16+65, color(160), fetch_font, 40, LEFT);
    textt(str(int(ValueGame[located[4]])), cak_x_icon+530,
    cak_y_icon16+65, color(102), fetch_value_font, 60, RIGHT);
  }
  if (max_t >= 32) {
    image(icon_Image[located[5]], cak_x_icon, cak_y_icon32, icon_size, icon_size);
    textt(Building_name[located[5]], cak_x_icon+100,
    cak_y_icon32+65, color(160), fetch_font, 40, LEFT);
    textt("kWh", cak_x_icon+550, cak_y_icon32+65,
    color(160), fetch_font, 40, LEFT);
    textt(str(int(ValueGame[located[5]])), cak_x_icon+530,
    cak_y_icon32+65, color(102), fetch_value_font, 60, RIGHT);
  }

  if (max_t >= 64) {
    image(icon_Image[located[6]], cak_x_icon, cak_y_icon64, icon_size, icon_size);
    textt(Building_name[located[6]], cak_x_icon+100,
    cak_y_icon64+65, color(160), fetch_font, 40, LEFT);
    textt("kWh", cak_x_icon+550, cak_y_icon64+65, color(160), fetch_font, 40, LEFT);
    textt(str(int(ValueGame[located[6]])), cak_x_icon+530,
    cak_y_icon64+65, color(102), fetch_value_font, 60, RIGHT);
  }
  if (max_t >= 128) {
    image(icon_Image[located[7]], cak_x_icon, cak_y_icon128,
    icon_size, icon_size);
    textt(Building_name[located[7]], cak_x_icon+100,
    cak_y_icon128+65, color(160), fetch_font, 40, LEFT);
    textt("kWh", cak_x_icon+550, cak_y_icon128+65, color(160), fetch_font, 40, LEFT);
    textt(str(int(ValueGame[located[7]])), cak_x_icon+530,
    cak_y_icon128+65, color(102), fetch_value_font, 60, RIGHT);
  }
}

void cak_icondisp_gameover() {
  // display icon pyramid on the right margin of game board up to board numbered max_t
  image(icon_Image[located[0]], cak_x_icon, cak_y_icon1, icon_size, icon_size);
  image(icon_Image[located[1]], cak_x_icon, cak_y_icon2, icon_size, icon_size);
  image(icon_Image[located[2]], cak_x_icon, cak_y_icon4, icon_size, icon_size);
  image(icon_Image[located[3]], cak_x_icon, cak_y_icon8, icon_size, icon_size);
  image(icon_Image[located[4]], cak_x_icon, cak_y_icon16, icon_size, icon_size);
  image(icon_Image[located[5]], cak_x_icon, cak_y_icon32, icon_size, icon_size);
  image(icon_Image[located[6]], cak_x_icon, cak_y_icon64, icon_size, icon_size);
  image(icon_Image[located[7]], cak_x_icon, cak_y_icon128, icon_size, icon_size);
  textAlign(LEFT);
  textFont(fetch_font, 40);
  text(Building_name[located[0]], cak_x_icon+100, cak_y_icon1+65);
  text(Building_name[located[1]], cak_x_icon+100, cak_y_icon2+65);
  text(Building_name[located[2]], cak_x_icon+100, cak_y_icon4+65);
  text(Building_name[located[3]], cak_x_icon+100, cak_y_icon8+65);
  text(Building_name[located[4]], cak_x_icon+100, cak_y_icon16+65);
  text(Building_name[located[5]], cak_x_icon+100, cak_y_icon32+65);
  text(Building_name[located[6]], cak_x_icon+100, cak_y_icon64+65);
  text(Building_name[located[7]], cak_x_icon+100, cak_y_icon128+65);
  text("kWh", cak_x_icon+550, cak_y_icon1+65);
  text("kWh", cak_x_icon+550, cak_y_icon2+65);
  text("kWh", cak_x_icon+550, cak_y_icon4+65);
  text("kWh", cak_x_icon+550, cak_y_icon8+65);
```

```
  text("kWh", cak_x_icon+550, cak_y_icon16+65);
  text("kWh", cak_x_icon+550, cak_y_icon32+65);
  text("kWh", cak_x_icon+550, cak_y_icon64+65);
  text("kWh", cak_x_icon+550, cak_y_icon128+65);
  textAlign(RIGHT);
  textFont(fetch_value_font, 60);
  text(int(ValueGame[located[0]]), cak_x_icon+530, cak_y_icon1+65);
  text(int(ValueGame[located[1]]), cak_x_icon+530, cak_y_icon2+65);
  text(int(ValueGame[located[2]]), cak_x_icon+530, cak_y_icon4+65);
  text(int(ValueGame[located[3]]), cak_x_icon+530, cak_y_icon8+65);
  text(int(ValueGame[located[4]]), cak_x_icon+530, cak_y_icon16+65);
  text(int(ValueGame[located[5]]), cak_x_icon+530, cak_y_icon32+65);
  text(int(ValueGame[located[6]]), cak_x_icon+530, cak_y_icon64+65);
  text(int(ValueGame[located[7]]), cak_x_icon+530, cak_y_icon128+65);
  filter(GRAY);
}

//////////////////////////////
// check condition game over
boolean gameover() {
  int[] dx = {
    1, -1, 0, 0
  }
  , dy = {
    0, 0, 1, -1
  };
  boolean out = true;
  for (int i = 0; i < 4; i++) if (go(dy[i], dx[i], false) != null) out = false;
  return out;
}

int[][] go(int dy, int dx, boolean updatescore) {
  int[][] bak = new int[cak_numrows][cak_numrows];
  for (int j = 0; j < cak_numrows; j++)
  for (int i = 0; i<cak_numrows; i++) bak[j][i] = b[j][i];
  boolean moved = false;
  if (dx != 0 || dy != 0) {
    int d =  dx != 0 ? dx : dy;
    for (int perp = 0; perp < b.length; perp++)
      for (int tang = (d > 0 ? b.length - 2 : 1);
      tang != (d > 0 ? -1 : b.length);
      tang-=d) {
        int y = dx != 0 ? perp : tang, x = dx != 0 ? tang : perp, ty = y, tx = x;
        if (bak[y][x]==0) continue;
        for (int i= (dx != 0 ? x : y)+d;
        i!= (d > 0 ? b.length : -1);
        i+=d) {
          int r = dx != 0 ? y : i, c = dx != 0 ? i : x;
          if (bak[r][c] != 0 && bak[r][c] != bak[y][x]) break;
          if (dx != 0) tx = i;
          else ty = i;
        }
        if ( (dx != 0 && tx == x) || (dy != 0 && ty == y)) continue;
        else if (bak[ty][tx]==bak[y][x]) {
          bak[ty][tx] *= 2;
          if (updatescore) score += bak[ty][tx];
          moved = true;
        } else if ( (dx != 0 && tx != x) || (dy != 0 && ty != y)) {
```

```
          bak[ty][tx] = bak[y][x];
          moved = true;
        }
        if (moved) bak[y][x] = 0;
      }
  }
  return moved ? bak : null;
}

//////////////////////////////
// hand event of energy game
//
void checkMouseCanVas2(int hand_x_position, int hand_y_position) {
  mouseX = hand_x_position ;
  mouseY = hand_y_position ;

  if (dead == 0) {
    preecha_previousMouseX = lerp(preecha_previousMouseX, mouseX, 0.09);
    preecha_previousMouseY = lerp(preecha_previousMouseY, mouseY, 0.09);

    float current_time = millis();
    int dx, dy;

    if (current_time - trigger_time > 1500 ) {
      fill(0, 255, 0);
      ellipse(mouseX, mouseY, 20, 20);
      image(hand_black, mouseX, mouseY);

      dy = (preecha_previousMouseY-mouseY) >= 150 ? -1 :
     (preecha_previousMouseY-mouseY)<=-150 ? 1 : 0;
      dx =(preecha_previousMouseX-mouseX)>=150 ? -1 :
     (preecha_previousMouseX-mouseX)<=-150 ? 1 : 0);
      if ((dx != 0) || (dy != 0)) {
        noSwipe+=1;
        trigger_time = millis();
      }
    } else {
      fill(255, 0, 0);
      ellipse(mouseX, mouseY, 20, 20);
      image(hand_white, mouseX, mouseY);
      dy = 0;
      dx = 0;
    }
    int[][] newb = go(dy, dx, true);
    if (newb != null) {
      b = newb;
      spawn();
    }
    if (gameover()) {
      dead = 1;
    }
  }
}
```

ฉ โปรแกรมในส่วนแสดงผลของแคนวาส EE information (canvas3.pde)

```
void canVas3() {
```

```
  try {
  pushMatrix();
  translate(preecha_imageShift_xpos_canvas3, preecha_imageShift_ypos_canvas3+150);
  if(preecha_showPicLeft_canvas3 = (preecha_allImages_canvas3.size()) - 1;
  } else {
  preecha_showPicLeft_canvas3 = preecha_showPic_canvas3 - 1;
  }

  if(preecha_showPic_canvas3 == (preecha_allImages_canvas3.size()) - 1 ) {
  preecha_showPicRight_canvas3 = 0;
  } else {
  preecha_showPicRight_canvas3 = preecha_showPic_canvas3 + 1;
    }

  PImage picLeft = preecha_allImages_canvas3.get(preecha_showPicLeft_canvas3);
  PImage pic = preecha_allImages_canvas3.get(preecha_showPic_canvas3);
  PImage picRight = preecha_allImages_canvas3.get(preecha_showPicRight_canvas3);
  image (picLeft, -width, 0);
  image (pic, 0, 0);
  image (picRight, width, 0);
  }

  catch (Exception e) {
  preecha_allImages_canvas3 = new ArrayList<PImage>();
  preecha_checkImgfiles(preecha_pictureFolder_canvas3);
  }
  preecha_imageShift_xpos_canvas3 += (preecha_imageShift_Target_canvas3
  -preecha_imageShift_xpos_canvas3)*preecha_easingRatio_canvas3;
  popMatrix();
}

//////////////////////////
/// hand event
void checkMouseCanVas3(int hand_x_position, int hand_y_position) {
  mouseX = hand_x_position ;
  mouseY = hand_y_position ;

  preecha_previousMouseX = lerp(preecha_previousMouseX, mouseX, 0.09);
  float current_time = millis();
  int dx;

  if (current_time - trigger_time > 2500 ) {
    image(hand_black, mouseX, mouseY);


    dx=(preecha_previousMouseX-mouseX)>=150 ? -1
    : ((preecha_previousMouseX-mouseX)<=-150 ? 1 : 0);

    if (dx == -1) {
      noSwipe +=1;
      preecha_showPic_canvas3 = (preecha_showPic_canvas3+1)
      %(preecha_allImages_canvas3.size());
      preecha_imageShift_xpos_canvas3 = width;
    }
    if (dx == 1) {
      noSwipe +=1;
      if (preecha_showPic_canvas3 == 0) {
        preecha_showPic_canvas3 = (preecha_allImages_canvas3.size())-1;
```

```
      } else {
        preecha_showPic_canvas3 = (preecha_showPic_canvas3-1);
      }
      preecha_imageShift_xpos_canvas3 = -width;
    }
    if (dx != 0) {
      trigger_time = millis();
    }
  } else {
    image(hand_white, mouseX, mouseY);
    dx = 0;
  }
}
```

ช โปรแกรมในส่วนแสดงผลของแคนวาส alarm & alert (canvas4.pde)

```
void canVas4() {
  int startday   = (checkFromcalendar(1, month(), year()))+1;
  int weekend = floor((8-startday)/7)+floor((startday+day()-1)/7)
  +floor((startday+day()-2)/7);

  if ((weekend%2)==0) {
    saturday = weekend/2;
    sunday   = weekend/2;
  }
  else {
    saturday = floor(weekend/2)+1;
    sunday   = floor(weekend/2);
  }
  int weekday  = day()- weekend;
  String [] targetdepartment = split(tarketQuata[1], ",");
  String [] targetEEfl2      = split(tarketQuata[2], ",");
  String [] targetEEfl3      = split(tarketQuata[3], ",");
  String [] targetEEfl4      = split(tarketQuata[4], ",");
  String [] targetEEfl5      = split(tarketQuata[5], ",");
  String [] targetENG4fl12   = split(tarketQuata[6], ",");
  String [] targetENG4fl13   = split(tarketQuata[7], ",");
  String [] targetHV         = split(tarketQuata[8], ",");

  target = new int [8];
  target[0] = int((weekday*int(targetdepartment[1]))
  +(saturday*int(targetdepartment[2]))+(sunday*int(targetdepartment[3])));
  target[1] = int((weekday*int(targetEEfl2[1]))
  +(saturday*int(targetEEfl2[2]))+(sunday*int(targetEEfl2[3])));
  target[2] = int((weekday*int(targetEEfl3[1]))
  +(saturday*int(targetEEfl3[2]))+(sunday*int(targetEEfl3[3])));
  target[3] = int((weekday*int(targetEEfl4[1]))
  +(saturday*int(targetEEfl4[2]))+(sunday*int(targetEEfl4[3])));
  target[4] = int((weekday*int(targetEEfl5[1]))
  +(saturday*int(targetEEfl5[2]))+(sunday*int(targetEEfl5[3])));
  target[5] = int((weekday*int(targetENG4fl12[1]))
  +(saturday*int(targetENG4fl12[2]))+(sunday*int(targetENG4fl12[3])));
  target[6] = int((weekday*int(targetENG4fl13[1]))
  +(saturday*int(targetENG4fl13[2]))+(sunday*int(targetENG4fl13[3])));
  target[7] = int((weekday*int(targetHV[1]))
  +(saturday*int(targetHV[2]))+(sunday*int(targetHV[3])));
 // check percentage
```

```
percent = int(ValueGame[zone_index])*100/target[zone_index];

radiusOfEnergyConsumed = (percent * 800)/100;
if (radiusOfEnergyConsumed >= 800) {
  radiusOfEnergyConsumed = 800;
}
radius = lerp(radius, radiusOfEnergyConsumed, 0.05);
fontBD = lerp(fontBD, 100, 0.1);
percentFontRatio = lerp(percentFontRatio, 200, 0.05);

fill(51, 51, 255, 220);
rect(10, height-200, 450, 150, 30);

fill(100);
textFont(textBuilding, fontBD);
textAlign(LEFT, TOP);
text(Building_name_alarm[zone_index], 20, 180);

textFont(textwarn, 70);
textAlign(LEFT, BOTTOM);
text("Energy Limit Target", 0+20, height-210);
textAlign(RIGHT, BOTTOM);
text("Energy Consumed", width-20, height-210);

textFont(textwarn, 50);
textAlign(LEFT, BOTTOM);
text("accumulate daily target (1st - today) of this month",
0+10, height-10);
textAlign(RIGHT, BOTTOM);
text("accumulate daily consumption (1st - today) of this month",
width-10, height-10);

fill(255);
textFont(textwarn, 150);
textAlign(LEFT, BOTTOM);
text(target[zone_index], 0+30, height-70);

textFont(textwarn, 100);
textAlign(RIGHT, BOTTOM);
text("kWh", 460, height-80);
text("kWh", width-10, height-80);

fill(51, 51, 255, 200);
pushMatrix();
translate(width/2, (height+200)/2);
for (int i= 0; i<60; i++) {
  rotate(TWO_PI/60);
  ellipse(400, 0, 10, 10);
}
popMatrix();
//  ellipse(width/2, (height+200)/2, 800, 800);
noFill();

if ((percent >= 70)&&(percent<90)) {
  ellipseMode(CENTER);
  fill(252, 204, 13);
  noStroke();
  ellipse(width/2, (height+200)/2, radius, radius);
```

```
    rect(width-460, height-200, 450, 150, 30);
    fill(252, 223, 81);
    textFont(alarm, 100);
    textAlign(LEFT, TOP);
    text("WARNING !!", 10, 300);
  }
  if (percent >= 90) {
    ellipseMode(CENTER);
    fill(255, 0, 0, 200);
    noStroke();
    ellipse(width/2, (height+200)/2, radius, radius);
    rect(width-460, height-200, 450, 150, 30);
    fill(255, 0, 0);
    textFont(alarm, 100);
    textAlign(LEFT, TOP);
    text("!! ALERT !!", 10, 300);
  }
  if (percent < 70) {
    ellipseMode(CENTER);
    fill(0, 255, 0, 200);
    noStroke();
    ellipse(width/2, (height+200)/2, radius, radius);
    rect(width-460, height-200, 450, 150, 30);
    fill(0, 255, 0);
    textFont(alarm, 100);
    textAlign(LEFT, TOP);
    text("Normal :)", 10, 300);
  }
  if (percent >= 60) {
    fill(255);
    textFont(alarm, percentFontRatio);
    textAlign(CENTER, CENTER);
    text(percent+" %", width/2, (height+200)/2);
  }
  if (percent < 60) {
    fill(100);
    textFont(alarm, percentFontRatio);
    textAlign(CENTER, CENTER);
    text(percent+" %", width/2, (height+200)/2);
  }

  fill(255);
  textFont(textwarn, 150);
  textAlign(LEFT, BOTTOM);
  text(int(ValueGame[zone_index]), width-460+20, height-70);
  textFont(textwarn, 100);
  textAlign(RIGHT, BOTTOM);
  text("kWh", width-10, height-80);
}


//////////////////////////////
// hand control eveent
//
void checkMouseCanVas4(int hand_x_position, int hand_y_position) {
  mouseX = hand_x_position ;
  mouseY = hand_y_position ;
```

```
  preecha_previousMouseX = lerp(preecha_previousMouseX, mouseX, 0.09);
  float current_time = millis();
  int dx;

  if (current_time - trigger_time > 2500 ) {
    image(hand_black, mouseX, mouseY);

    dx=(preecha_previousMouseX-mouseX)>=150 ? -1
    : ((preecha_previousMouseX-mouseX)<=-150 ? 1 : 0);

    if (dx == -1) {
      noSwipe +=1;
      zone_index = (zone_index+1)%8;
    }
    if (dx == 1) {
      noSwipe +=1;
      if (zone_index == 0) {
        zone_index = 7;
      }
      else {
        zone_index = (zone_index-1);
      }
    }
    if (dx != 0) {
      trigger_time = millis();
      fontBD = 0;
      percentFontRatio = 0;
    }
  }
  else {
    image(hand_white, mouseX, mouseY);
    dx = 0;
  }
}
```

ซ โปรแกรมในส่วนแสดงผล menu เพื่อใช้เลือกแคนวาส (canvasMenu.pde)

```
void canVas_Menu() {
  background(255);
  imageMode(CORNER);
  clock(1680, 85, 1680, 140);

  if (overPicture(((0+360)/2)-100, 0, 100, menuHeight_margin)) {
    overPicture1 = true;
    image(canvas_Selection1over, 0, 10);
    image(canvas_Selection2, 360, 6);
    image(canvas_Selection3, 720, 8);
    image(canvas_Selection4, 1080, 8);
  } else if (overPicture(((360+720)/2)-100, 0, 100, menuHeight_margin)) {
    overPicture2 = true;
    image(canvas_Selection1, 0, 10);
    image(canvas_Selection2over, 360, 6);
    image(canvas_Selection3, 720, 8);
    image(canvas_Selection4, 1080, 8);
  } else if (overPicture(((720+1080)/2)-100, 0, 100, menuHeight_margin)) {
    overPicture3 = true;
    image(canvas_Selection1, 0, 10);
```

```
  image(canvas_Selection2, 360, 6);
  image(canvas_Selection3over, 720, 8);
  image(canvas_Selection4, 1080, 8);
} else if (overPicture(((1080+1440)/2)-100, 0, 100, menuHeight_margin)) {
  overPicture4 = true;
  image(canvas_Selection1, 0, 10);
  image(canvas_Selection2, 360, 6);
  image(canvas_Selection3, 720, 8);
  image(canvas_Selection4over, 1080, 8);
} else {
  overPicture1 = overPicture2 = overPicture3 = overPicture4 = false;
  image(canvas_Selection1, 0, 10);
  image(canvas_Selection2, 360, 6);
  image(canvas_Selection3, 720, 8);
  image(canvas_Selection4, 1080, 8);
}

////////////////////////////////////////////////
// now hand is NOT over any pictures of menu icon
//

if (!(overPicture1)&& !(overPicture2)&& !(overPicture3)&& !(overPicture4)) {

  // assign the most recent time that hand is not over any menu icon //
  // pictures to variable <preecha_resetSelect_icon>                 //

  preecha_resetSelect_icon = millis();

  // assign temporary variable //

}

////////////////////////////////
// now hand is over Picture 1
//

if (overPicture1) {

  // compute how long the hand has touched this picture //
  int preecha_passTimecanvasMenu = millis() - preecha_resetSelect_icon;

  // check if the hand has touched this picture for long enough time //
  if (preecha_passTimecanvasMenu >= preecha_timeovercanvas_selected) {

    // now hand has touched this picture long enough //
    // let us now activate the canvas                //

    // assign canvas number according to the touched picture //
    canvas_number = 1;

    if (alreadyRunInitialisationCodeInCurrentCanvas == 0) {

      // run the following canvas-initialisation code only //
      // once per each time of entering this canvas        //

      alreadyRunInitialisationCodeInCurrentCanvas = 1;
      //        canvas1_init();
    }
```

```
  }
}

////////////////////////////////
// now hand is over Picture 2
//

if (overPicture2) {

  // compute how long the hand has touched this picture //
  int preecha_passTimecanvasMenu = millis() - preecha_resetSelect_icon;

  // check if the hand has touched this picture for long enough time //
  if (preecha_passTimecanvasMenu >= preecha_timeovercanvas_selected) {

    // now hand has touched this picture long enough //
    // let us now activate the canvas                //

    // assign canvas number according to the touched picture //
    canvas_number = 2;

    if (alreadyRunInitialisationCodeInCurrentCanvas == 0) {

      // run the following canvas-initialisation code only //
      // once per each time of entering this canvas        //

      alreadyRunInitialisationCodeInCurrentCanvas = 1;
      canvas2_init();
    }
  }
}


////////////////////////////////
// now hand is over Picture 3
//

if (overPicture3) {

  // compute how long the hand has touched this picture //
  int preecha_passTimecanvasMenu = millis() - preecha_resetSelect_icon;

  // check if the hand has touched this picture for long enough time //
  if (preecha_passTimecanvasMenu >= preecha_timeovercanvas_selected) {

    // now hand has touched this picture long enough //
    // let us now activate the canvas                //

    // assign canvas number according to the touched picture //
    canvas_number = 3;

    if (alreadyRunInitialisationCodeInCurrentCanvas == 0) {

      // run the following canvas-initialisation code only //
      // once per each time of entering this canvas        //
      //        canvas3_init();
      alreadyRunInitialisationCodeInCurrentCanvas = 1;
    }
```

```
    }
  }

  ////////////////////////////////
  // now hand is over Picture 4
  //

  if (overPicture4) {

    // compute how long the hand has touched this picture //
    int preecha_passTimecanvasMenu = millis() - preecha_resetSelect_icon;

    // check if the hand has touched this picture for long enough time //
    if (preecha_passTimecanvasMenu >= preecha_timeovercanvas_selected) {

      // now hand has touched this picture long enough //
      // let us now activate the canvas                //

      // assign canvas number according to the touched picture //
      canvas_number = 4;

      if (alreadyRunInitialisationCodeInCurrentCanvas == 0) {

        // run the following canvas-initialisation code only //
        // once per each time of entering this canvas        //
        //        canvas4_init();
        alreadyRunInitialisationCodeInCurrentCanvas = 1;
      }
    }
  }
}


boolean overPicture(int x, int y, int sizeX, int sizeY) {
  if (mouseX >= x && mouseX <= x+sizeX && mouseY >= y && mouseY <= y+sizeY) {
    return true;
  } else {
    return false;
  }
}
```

ฌ โปรแกรมในส่วนของการโหลดรูปภาพจากโฟลเดอร์เพื่อใช้ในแคนวาสต่าง ๆ (checkImagefile.pde)

```
void preecha_checkImgfiles(String preecha_pictureFolder) {
  String[] outList_of_foundImageFiles = {};
  String[] list_of_imageFileSuffixes = {"jpg", "png"};

  File file = new File(preecha_pictureFolder);
  String[] names = file.list();

  for (int numberImage =0; numberImage < names.length; numberImage++ ) {
    String[] curr_filenameSplit = splitTokens( names[numberImage], "." );

   for (int fileSuffix_i = 0; fileSuffix_i < list_of_imageFileSuffixes.length; fileSuffix_i++){
```

```
      String examinedFile_filesuffix = curr_filenameSplit[1] ;
      String listOfValid_fileSuffixed = list_of_imageFileSuffixes[fileSuffix_i] ;

      if ( examinedFile_filesuffix.equals( listOfValid_fileSuffixed ) ) {
       outList_of_foundImageFiles =
       append( outList_of_foundImageFiles, preecha_pictureFolder+names[numberImage] );
      }
    }
  }

  ////////////////////////////////////
  ///  loading picture of each canvas
  //

  for (int i =0; i < outList_of_foundImageFiles.length; i++) {
    if (preecha_pictureFolder == preecha_pictureFolder_canvas1) {
    preecha_allImages.add(loadImage(outList_of_foundImageFiles[i]));
   (preecha_allImages.get(i)).resize(width, height - menuHeight_margin);
    }
    if (preecha_pictureFolder == preecha_pictureFolder_canvas3) {
    preecha_allImages_canvas3.add(loadImage(outList_of_foundImageFiles[i]));
   (preecha_allImages_canvas3.get(i)).resize(width, height - menuHeight_margin);
    }
    if (preecha_pictureFolder == preecha_pictureFolder_canvas0) {
    preecha_allImages_canvas0.add(loadImage(outList_of_foundImageFiles[i]));
   (preecha_allImages_canvas0.get(i)).resize(640, 400);
    }
  }
}
```

ฒ โปรแกรมในส่วนแสดงผลนาฬิกา และปฏิทิน (clock.pde)

```
void clock(int clock_x, int clock_y, int calendar_x, int calendar_y ){

  String[] dayName = {
  "Sunday", "Monday", "Tuesday", "Wednesday",
  "Thursday", "Friday", "Saturday"
  };
  String[] monthName = {
  "January", "February", "March", "April",
  "May", "June", "July", "August",
  "September", "October", "November", "December"
  };

  // Draw current time (Draw Hour/Minute/AM or PM)
  day = day();    // Values from 1 - 31
  month = month();  // Values from 1 - 12
  year = year();   // 2003, 2004, 2005, etc.
  sec = second();
  min = minute();
  hour = hour();

  int christYEAR = year - 543;
  textFont(clock_font, 150);
  textAlign(CENTER);

  // Define Minute (fixes 10 minute bug)
```

```
  if (min<10) {
    fill(102, 102, 102, 70);
    noStroke();
    rect(1440, 0, width, 150);
    noFill();
    fill(102, 102, 102);
    text(hour+":0"+min, clock_x, clock_y);
    textFont(calendar_font,40);
    textAlign(CENTER);
    text(dayName[weekNumber]+"  "+day+"  "+
    monthName[month-1]+"  "+year, calendar_x, calendar_y);
  }
  else {
    fill(102, 102, 102, 70);
    noStroke();
    rect(1440, 0, width, 150);
    noFill();
    fill(102, 102, 102);
    text(hour+":"+min, clock_x, clock_y);
    textFont(calendar_font,40);
    textAlign(CENTER);
    text(dayName[weekNumber]+"  "+day+"  "+
    monthName[month-1]+"  "+year, calendar_x, calendar_y);
  }
  weekNumber = checkFromcalendar(day, month, year);
}

// to check what day is it
int checkFromcalendar(int d, int m, int y){
  if (m < 3) {
    m += 12;
    y--;
  }
  return(d+int((m+1)*2.6)+y+int(y/4)+6*int(y/100)+int(y/400)+6)%7;
}
```

ด โปรแกรมในส่วนของการตรวจจับ และติดตามโครงกระดูกของผู้ใช้งาน (kinect.pde)

```
void trackingUser() {
  kinect.update();
  kinect.getUsers();
  preecha_getVisibleUsers();

  if (userList.length == 0) {
  //println("Nouser");
  } else {
  //printArray(userList);
  }

  if (userList.length > 0) {
   if (userList.length == 1) {
    controlUserId = userList[0];
    }
    if (kinect.isTrackingSkeleton(controlUserId)) {
    kinect.getJointPositionSkeleton(controlUserId, SimpleOpenNI.SKEL_RIGHT_HAND, rightHand);
```

```
      kinect.convertRealWorldToProjective(rightHand, convertedrightHand);
      xMapped = map(convertedrightHand.x, 0, 640, 0, width);
      yMapped = map(convertedrightHand.y, 0, 480, 0, height);
     }
  }
}

//////////////////////////
// SimpleOpenNI event
//

void onNewUser(SimpleOpenNI curContext, int userId) {
  curContext.startTrackingSkeleton(userId);
  userCount += 1;
}

void preecha_getVisibleUsers() {
  userList = new int[0];
  for (int i = 1; i <= 6; i++) {
   if (60*60*hour()+60*minute()+second()-userLastTimeSeen[i-1]<0.2){
     userList = append(userList, i);
    }
  }
}

void onVisibleUser(SimpleOpenNI curContext, int userId) {
  userLastTimeSeen[userId-1] = 60*60*hour()+60*minute()+second();
}

void onLostUser(SimpleOpenNI curContext, int userId) {
  userCount -= 1;
  if (userId == controlUserId) {
    controlUserId = userList[0];
  }
}

float detect_hand_x_position() {return xMapped;}
float detect_hand_y_position() {return yMapped;}
```

ประวัติผู้เขียนวิทยานิพนธ์

ปรีชา ขาวสะอาด เกิดเมื่อวันที่ 23 พฤษภาคม พ.ศ. 2528 ที่จังหวัดกรุงเทพฯ สำเร็จการศึกษาชั้นมัธยมศึกษาจากโรงเรียนเทพศิรินทร์ ในปีการศึกษา 2545 จากนั้นได้เข้าศึกษาต่อที่คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมอิเล็คทรอนิกส์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง จนสำเร็จหลักสูตรวิศวกรรมศาสตรบัณฑิตในปีการศึกษา 2550 หลังจากจบการศึกษาได้เข้าทำงานเป็นวิศวกรฝ่ายขายที่บริษัทอิเล็คทรอนิคซอร์สในปี 2550 จนถึงปี 2553 จากนั้นเข้าทำงานเป็นวิศวกรโรงงานในบริษัทสีลมการแพทย์ในปี 2553 และในปี 2554 ย้ายตำแหน่งเป็นวิศวกรฝ่ายผลิตจนถึงปี 2555 จากนั้นได้เข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ณ จุฬาลงกรณ์มหาวิทยาลัย จนสำเร็จปีการศึกษาในปีการศึกษา 2557

บทความทางวิชาการจากวิทยานิพนธ์

[1] P. Khawsa-ard and C. Aswakul. Application of Simple Computer Board Game with Gesture Sensor Input for Increasing Awareness in Electrical Energy Consumption. 29th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2014), Phuket, Thailand, July 2014.