ระบบการระบุตัวตนโดยใช้โครงกระดูกร่างกาย

Human Identification by Using Skeleton

พัทธ์ธีรา งามเอก
Phatheera Ngam-ake

ธนพัตร เกาะแก้ว
Thanaphat Kho-kaew

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2559

ระบบการระบุตัวตนโดยใช้โครงกระดูกร่างกาย

Human Identification by Using Skeleton

พัทธ์ธีรา งามเอก

Phatheera Ngam-ake

ธนพัตร เกาะแก้ว

Thanaphat Kho-kaew

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบการระบุตัวตนโดยใช้โครงกระดูกร่างกาย

Human Identification by Using Skeleton

โดย

พัทธ์ธีรา งามเอก

ธนพัตร เกาะแก้ว

อาจารย์ที่ปรึกษา

ผศ.ดร.ภัทรพงษ์ ผาสุขกิจ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต ปีการศึกษา 2559

ภาควิชา	วิศวกรรมอิเล็กทรอนิกส์	
คณะ	วิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	
เรื่อง	ระบบการระบุตัวตนโดยใช้โครงกระดูกร่างกาย Human Identification by Using Skeleton	
ผู้จัดทำ	นางสาวพัทธ์ธีรา งามเอก	รหัสประจำตัว 56010489
	นายธนพัตร เกาะแก้ว	รหัสประจำตัว 56010534

รายงานนี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

ภ.ผ.

(ผศ.ดร.ภัทรพงษ์ ผาสุขกิจ)
อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ระบบการระบุตัวตนโดยใช้โครงกระดูกร่างกาย
นักศึกษา	นางสาวพัทธ์ธีรา งามเอก รหัสประจำตัว 56010489
	นายธนพัตร เกาะแก้ว รหัสประจำตัว 56010534
ปริญญา	วิศวกรรมศาสตรบัณฑิต
สาขาวิชา	วิศวกรรมอิเล็กทรอนิกส์
ปีการศึกษา	2559
อาจารย์ที่ปรึกษา	ผศ.ดร.ภัทรพงษ์ ผาสุขกิจ

บทคัดย่อ

โครงงานนี้นำเสนอเกี่ยวกับการพัฒนาระบบตรวจสอบตัวตน เพื่อเพิ่มความปลอดภัยให้แก่ระบบ โดยนำกล้อง Kinect One มาประยุกต์ใช้ในการระบุตัวตน ซึ่งสามารถแบ่งระบบได้เป็น 2 ส่วน ได้แก่ ส่วนแรก คือ ระบบการตรวจจับข้อต่อและทำการประมวลผลเข้ากับโปรแกรมเพื่อคำนวณความยาวส่วนต่างๆ ของร่างกาย และส่วนที่สอง คือ การเชื่อมต่อเข้ากับระบบกลอนประตูชนิดแม่เหล็กในระบบรักษาความปลอดภัย ซึ่งได้ทำการประยุกต์ใช้ไมโครคอนโทรลเลอร์ชนิด Arduino และ Relay ในการออกแบบ เพื่อให้สามารถสั่งงานได้สะดวกยิ่งขึ้น โดยระบบระบุตัวตนโดยใช้โครงกระดูกร่างกายนี้ จะช่วยให้องค์กรต่างๆ มีความปลอดภัยและมีเสถียรภาพยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Human Identification by Using Skeleton
Students	Miss Phatheera Ngam-ake Student ID 56010489
	Mr. Thanaphat Kho-kaew Student ID 56010534
Degree	Bachelor of Engineering
Program	Electronics Engineering
Year	2016
Project Advisor	Asst. Prof. Dr. Pattarapong Phasukkit

ABSTRACT

This project is presented about new ways to develop the identification system to improve the organization's security by using Kinect One. The new system can be separated into 2 parts. First, the system that designed to detect human's joints and evaluate with the identification system program to calculate the body part's lengths. The user's data will be kept in the organization database to be used in the next system. Second, this project also connected with magnetic lock in the security system that apply from microcontroller "Arduino" and relay to design the system functions and make the system easy to use. With this project, the organization will be more security and more stability.

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การนำ Kinect มาประยุกต์ใช้ในการระบุตัวตนบุคคลนี้ สามารถสำเร็จลุล่วงตามวัตถุประสงค์ คณะผู้จัดทำต้องขอขอบคุณ ผศ.ดร.ภัทรพงษ์ ผาสุขกิจ อาจารย์ประจำภาควิชาวิศวกรรมอิเล็กทรอนิกส์ ที่ได้ให้คำแนะนำและช่วยเหลือจนทำให้โครงงานชิ้นนี้สามารถสำเร็จได้ตามจุดประสงค์ ขอขอบคุณอาจารย์ท่านอื่นๆ และรุ่นพี่ ที่การเสียสละเวลามาดูแลในระหว่างการทดลองตลอดจนให้คำปรึกษาต่างๆ ขอขอบคุณเพื่อนๆ ที่ให้ความร่วมมือในระหว่างทำการทดสอบ ช่วยกันแนะนำ และแก้ไขปัญหาต่างๆ ไปพร้อมๆ กัน

สุดท้ายนี้ข้าพเจ้าขอขอบคุณในการช่วยเหลือของทุกท่านที่ทำให้โครงงานนี้สำเร็จลุล่วงมาได้ด้วยดี หากผิดพลาดพลาดประการใด ต้องขออภัยมา ณ ที่นี้ด้วย

พัทธ์ธีรา งามเอก

ธนพัตร เกาะแก้ว

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การปกป้องความมั่นคงปลอดภัยของระบบและข้อมูลภายในองค์กรถือเป็นเรื่องสำคัญ เนื่องจากในปัจจุบันมักเกิดการคุกคามโดยผู้ไม่ประสงค์ดี หรือจากโปรแกรมบางประเภทได้เพิ่มมากขึ้นและอาจนำมาซึ่งความเสียหายอย่างมากต่อองค์กร ดังนั้น ถ้าภายในระบบมีการควบคุมความปลอดภัยที่ดีจะช่วยลดโอกาสเสี่ยงต่อการถูกคุกคามได้

กระบวนการการพิสูจน์ตัวตนจะนำหลักฐานที่ผู้ใช้กล่าวอ้างมาตรวจสอบว่าบุคคลที่กล่าวอ้างนั้นเป็นใครและได้รับอนุญาตให้สามารถเข้ามาภายในระบบ ได้หรือไม่ การพิสูจน์ตัวตนมีหลายประเภทที่ใช้อยู่ในปัจจุบัน เช่น การพิสูจน์ตัวตนโดยใช้รหัสผ่าน ลักษณะเฉพาะทางชีวภาพของแต่ละบุคคล หรือ โดยใช้รหัสผ่านที่ใช้เพียงครั้งเดียว เป็นต้น แต่ละชนิดนั้นจะมีข้อดีข้อเสียแตกต่างกันไปขึ้นอยู่กับความจำเป็นในการใช้งาน

ในโครงงานนี้ การระบุตัวตนของบุคคลจะดำเนินการโดยค้นหาตำแหน่งของข้อต่อบนร่างกายมนุษย์ด้วยกล้อง Kinect One เพื่อนำไปประมวลผลเป็นค่าความยาวของส่วนต่างๆ ของร่างกาย โดยค่าที่คำนวณได้จะได้รับการบันทึกไว้ในฐานข้อมูลของระบบ เพื่อใช้ในการสั่งงานไปยังระบบเทียบค่าซึ่งเชื่อมต่อกับกลอนประตูชนิดแม่เหล็กที่ได้ประยุกต์ใช้งานบอร์ดไมโครคอนโทรลเลอร์ ชนิด Arduino และรีเลย์ร่วมกัน โดยระบบระบุตัวตนนี้จะสามารถแก้ปัญหาต่างๆ ซึ่งอาจเกิดจากระบบระบุตัวตนประเภทอื่นๆ ได้ เช่น ลดโอกาสในการส่งผ่านเชื้อโรคระหว่างบุคคล หรือปัญหาลายนิ้วมือผิดเพี้ยน เป็นต้น

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1. เพื่อศึกษาการทำงานและการใช้งานของ Kinect ในการตรวจจับร่างกายมนุษย์
2. เพื่อศึกษาซอฟต์แวร์ที่ใช้ในการเขียนโปรแกรม ได้แก่ Microsoft Visual Studio และ Arduino
3. สามารถหาจุดบกพร่องต่างๆ และสามารถแก้ไขปัญหาที่อาจเกิดขึ้นได้ในระหว่างทำการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 สมมติฐานของการศึกษา

นำ Kinect มาประยุกต์ใช้ในการตรวจจับลักษณะร่างกายของมนุษย์ และทำการประมวลผลเป็นค่าความยาวของส่วนต่างๆ ของร่างกายด้วยโปรแกรม Microsoft Visual Studio จากนั้นทำการเปรียบเทียบค่าด้วยโปรแกรม XAMPP เพื่อสั่งการไปยังวงจร Arduino ซึ่งเชื่อมต่อกับวงจรเปิด-ปิดประตู ให้อนุญาตให้ผู้ใช้งานเข้าสู่ภายในได้

1.4 ขอบเขตการวิจัย

1. ออกแบบโปรแกรมที่ใช้ในการระบุตัวตนด้วยกล้อง Kinect One
2. ทำการทดสอบกับกลุ่มตัวอย่างทั้งหมด 10 คน แบ่งเป็นผู้ชาย 8 คน และผู้หญิง 2 คน
3. บันทึกค่าความยาวของส่วนต่างๆ ในร่างกาย โดยมีตำแหน่งยืนห่างจากกล้อง 2.15 เมตร
4. ในการบันทึกค่าความยาวของร่างกาย กลุ่มตัวอย่างต้องยืนลำตัวตรง แยกแขนออกจากลำตัวเล็กน้อย
5. นำผลการทดสอบที่ได้มาวิเคราะห์ หากผลที่ได้ไม่ตรงตามที่ต้องการ จึงนำผลการทดสอบที่ได้ไปปรับปรุงแก้ไขโปรแกรมต่อไป
6. เมื่อโปรแกรมที่ทดสอบสามารถทำงานได้ตามต้องการแล้ว เริ่มทดสอบการระบุตัวตน โดยให้กลุ่มตัวอย่างยืนในตำแหน่งเดิมที่ทำการบันทึกค่า

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. เข้าใจหลักการทำงานของกล้อง Kinect รวมทั้งสามารถออกแบบโปรแกรมเพื่อใช้ในการบันทึกค่าจากกล้องได้
2. สามารถหาจุดบกพร่อง และสาเหตุต่างๆ ที่ทำให้ผลการทดลองมีการคลาดเคลื่อน และสามารถ ปรับปรุงแก้ไขวิธีการทดสอบเพื่อให้ได้ผลตรงตามที่ต้องการ
3. สามารถออกแบบโปรแกรมสำหรับระบุตัวตน เพื่อเพิ่มความปลอดภัยให้กับระบบ ผ่านการประยุกต์ใช้งานกล้อง Kinect

1.6 โครงสร้างรายงาน

ระบบตรวจจับลักษณะร่างกายของมนุษย์ด้วย Kinect ได้อธิบายถึงขั้นตอนและวิธีการออกแบบ โดยส่วนเนื้อหาของบทต่างๆ ดังต่อไปนี้

บทที่ 1 บทนำ ในบทนี้จะกล่าวถึง ความเป็นมาและความสำคัญ รวมถึงวัตถุประสงค์ของการศึกษา ขอบเขตของการศึกษา และประโยชน์ที่คาดว่าจะได้รับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง ในบทนี้จะกล่าวถึง ความรู้ที่เกี่ยวกับระบบตรวจจับการเคลื่อนไหวแบบ 3 มิติ หรือ Kinect การใช้งานโปรแกรม Microsoft Visual Studio, XAMPP และ Arduino

บทที่ 3 การออกแบบโปรแกรมที่ใช้ในการทดสอบ โดยจะกล่าวถึงซอฟต์แวร์ การออกแบบสถานที่ การทำการทดสอบ และการออกแบบการทดสอบ

บทที่ 4 ผลการทดลอง ในบทนี้จะกล่าวถึงผลการทดลองวัดค่าความยาวของส่วนต่างๆ ในร่างกาย ได้แก่ หัวไหล่ ข้อศอก ข้อมือ ศีรษะและเท้า ซึ่งสามารถแบ่งการทดลองออกได้เป็น 3 การทดลอง คือ การวัดสัดส่วนที่ได้จาก Kinect และผลการทดสอบระบบระบุตัวตนเมื่อทำการเชื่อมต่อกับบอร์ด Arduino

บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ ในบทนี้จะกล่าวถึง สรุปผลการวิจัย และข้อเสนอแนะ รวมไปถึงแนวทางในการพัฒนาต่อไป

1.7 แผนการปฏิบัติงาน

หัวข้อการปฏิบัติงาน / ช่วงเวลา(เดือน/วันที่)	ส.ค. 2559	ก.ย. 2559	ต.ค. 2559	พ.ย. 2559	ธ.ค. 2559	ม.ค. 2560	ก.พ. 2560	มี.ค. 2560
	26-31	1-30	1-31	1-30	1-31	1-31	1-28	1-31
1. ศึกษาซอฟต์แวร์								
2. ทำการพัฒนาโปรแกรมเพื่อให้มีความถูกต้องมากยิ่งขึ้น								
3. ดำเนินการทดสอบโปรแกรม และดำเนินการแก้ไขหากพบข้อผิดพลาด								

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 Kinect

Kinect เป็นอุปกรณ์เสริมของเครื่องเล่นเกม Xbox ที่ถูกผลิตมาเพื่อเพิ่มความสมจริงและความ สนุกสนานในการเล่นเกมโดยใช้เทคโนโลยีที่วิจัยและพัฒนาร่วมกับ PrimeSensor เป็นตัวขับเคลื่อน ในช่วงแรกนั้น ทางไมโครซอฟต์ซึ่งเป็นผู้ผลิตได้ใช้ชื่ออย่างไม่เป็นทางการว่าโปรเจคนาทาล (Project- Natal) จนกระทั่งเปลี่ยนมาใช้ชื่อ Kinect ซึ่งมาจากคำว่าไคเนติก (Kinetic) รวมกับคำว่าคอนเนค (Connect)

การทำงานของ Kinect เมื่อเริ่มทำงาน ผู้เล่นจะต้องสมมติตัวเองเป็นจอยคอนโทรลเลอร์ (Joy Controller) กล้องจะรับภาพการเคลื่อนไหวของผู้เล่นจากกล้องเข้าสู่ชิพประมวลและทำการแปลงสัญญาณออกมา โดยทุกเกมที่ใช้อุปกรณ์ Kinect จึงต้องใช้ร่างกายของผู้เล่นทุกส่วน เช่น เมื่อใช้กับเกมตีปิงปอง ผู้เล่นเพียงแค่ขยับร่างกายในลักษณะของการตีปิงปอง โดยไม่ต้องถือจอยสติก ทำให้ผู้เล่นมีความรู้สึกสมจริง และมีส่วนร่วมกับการเล่นเกมมากยิ่งขึ้น

2.1.1 Kinect 360

รูปที่ 2.1 อุปกรณ์ Kinect และเครื่องเล่นเกม Xbox

ที่มา : http://g-ecx.images-amazon.com/images/G/02/uk-ideogames/2010/Xbox

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

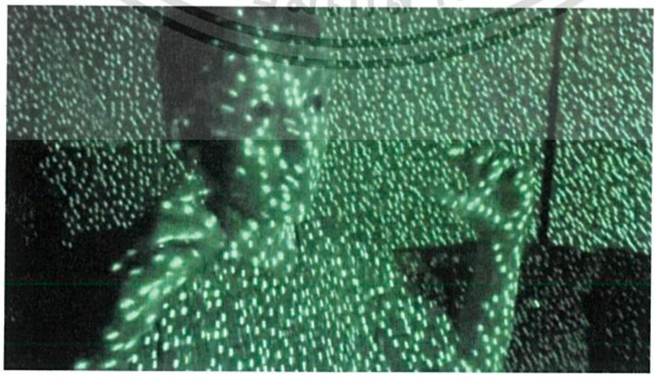
2.1.1.1 ส่วนประกอบและคุณสมบัติของ Kinect 360

รูปที่ 2.2 ส่วนประกอบของ Kinect 360

ที่มา : https://mcu56.learninginventions.org/wp-content/uploads/2014/01/sp-kinect-img1.png

1) Infrared Optics (3D Depth Sensors)

640x480 pixels 11-bit ความลึกระหว่าง 1.2 – 3.5 เมตร ค่าขอบเขตการมองเห็น (Field Of View) แนวตั้ง 43 องศา แนวนอน 57 องศา ใช้ในการตรวจจับตำแหน่งของผู้เล่น โดยสามารถวิเคราะห์ได้ทั้ง 3แกน ได้แก่ กว้าง ยาว ระยะใกล้ - ไกล ได้มากถึง 48 จุด พร้อมกัน 2คน ประกอบด้วย 2 ส่วน ได้แก่ *Infrared Laser Projector* ทำหน้าที่จะส่งแสงอินฟราเรดไปกระทบกับวัตถุที่อยู่ในแนวรัศมีและ *Monochrome CMOS Sensor* ทำหน้าที่เป็นตัวรับ หลังจากแสงอินฟราเรดตกกระทบกับวัตถุแล้ว



รูปที่ 2.3 แสงอินฟราเรดที่ใช้ในการหาความลึกของวัตถุ

ที่มา : https://i.ytimg.com/vi/0Yr5_0imcbY/maxresdefault.jpg

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) RGB Camera 640 x 480 pixels 8-bit 30 fps

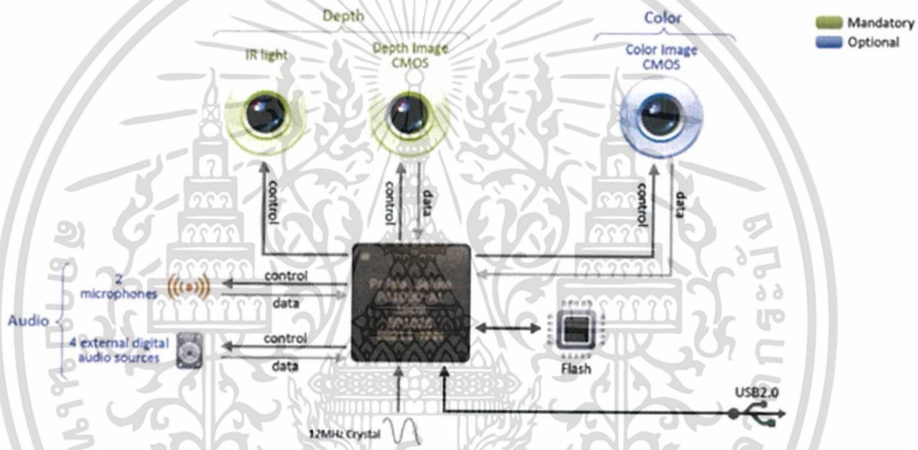
เป็นกล้องถ่ายภาพเคลื่อนไหวซึ่งช่วยในระบบจดจำใบหน้า (Face recognition) และการตรวจจับลักษณะเด่นอื่นๆ โดยใช้ความสามารถในการตรวจจับองค์ประกอบของสีทั้งสามสี ได้แก่ สีแดง สีเขียว และสีน้ำเงิน

3) Motorized Tilt

ทำหน้าที่ปรับมุมมองตัวกล้อง โดยสามารถปรับขึ้น-ลง ได้ 27 องศา รวมทั้งหมด 54 องศา

4) Multi-Array Microphone จำนวน 4 ตัว 16-bit 16 kHz

ทำหน้าที่ใช้รับเสียงจากผู้เล่น สามารถใช้แยกแยะเสียงของผู้เล่นและเสียงรบกวนที่อยู่บริเวณรอบๆ ได้ รวมทั้งสามารถระบุตำแหน่งของผู้ใช้ได้ ว่าอยู่ตำแหน่งใด



รูปที่ 2.4 สถาปัตยกรรมของ Kinect

ที่มา : http://www.ayarafun.com/wp-content/uploads/2010/11/kinect_system.jpg

ตารางที่ 2.1 คุณสมบัติของ Kinect 360

Property	Spec
Field of View (Horizontal, Vertical, Diagonal)	58° H, 45° V, 70° D
Depth image size	VGA (640x480)
Spatial x/y resolution (@ 2m distance from sensor)	3mm
Depth z resolution (@ 2m distance from sensor)	1cm
Maximum image throughput (frame rate)	60fps
Operation range	0.8m – 3.5m
Color image size	UXGA (1600x1200)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Audio: built-in microphones	Two microphones
Audio: digital inputs	Four inputs
Data interface	USB 2.0
Power supply	USB 2.0
Power consumption	2.25W
Dimensions (Width x Height x Depth)	14cm x 3.5cm x 5cm
Operation environment (every lighting condition)	Indoor
Operating temperature	0°C – 40°C

2.1.2 Kinect ONE

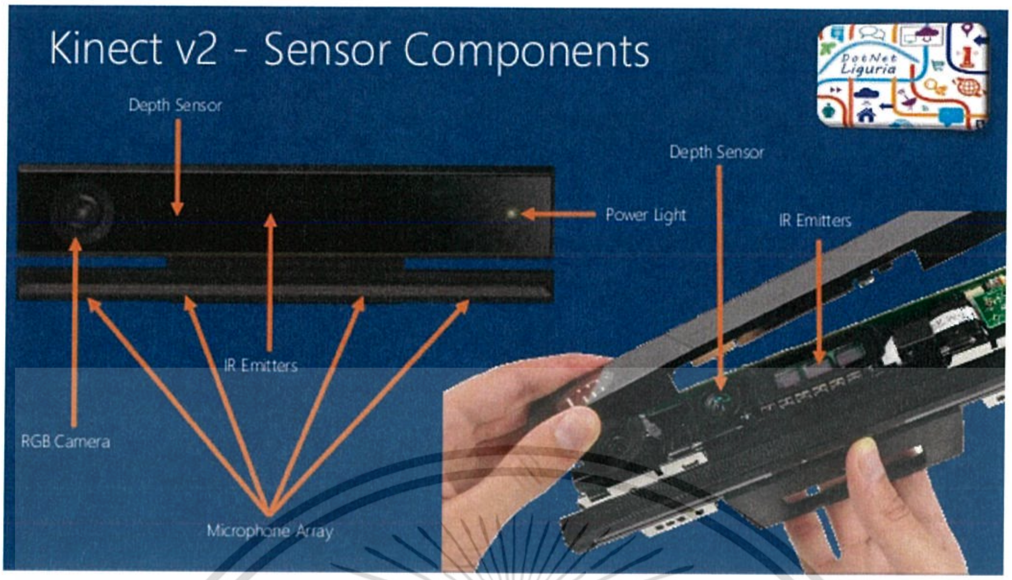
Kinect One เป็นคอนโซลวิดีโอเกมที่พัฒนาขึ้นโดย Microsoft เพื่อทดแทน Kinect 360 โดย Xbox One จะให้ความสำคัญกับระบบ Cloud Computing รวมถึงการผสานรวมแอพพลิเคชั่นและบริการด้านความบันเทิงเข้าไว้ด้วยกัน ซึ่งจะช่วยให้สามารถซ้อนทับรายการโทรทัศน์สดจากกล่อง set-top box หรือจูนเนอร์ดิจิตอลสำหรับทีวีภาคพื้นดินได้ นอกจากนี้ Kinect One ยังประกอบด้วยเซ็นเซอร์ "Kinect 2.0" ซึ่งจะช่วยให้การติดตามการเคลื่อนไหวและการจดจำเสียงดีขึ้น และผู้ใช้งานยังสามารถบันทึกและแชร์คลิปจากเพลย์หรือสตรีมสดไปยัง Twitch, Ustream และแพลตฟอร์มสตรีมมิ่งอื่นๆ ได้อีกด้วย อีกทั้งยังสามารถตรวจจับอัตราการเต้นของหัวใจ การแสดงออกทางสีหน้า และการวางแนวของข้อต่อต่างๆได้ถึง 25 ข้อ (รวมทั้งนิ้วมือ) น้ำหนักที่วางอยู่บนแขนขา ความเร็วของการเคลื่อนไหวของผู้เล่นและท่าทางต่างๆ ด้วยตัวควบคุมมาตรฐาน

รูปที่ 2.5 อุปกรณ์ Kinect และเครื่องเล่นเกม Xbox

ที่มา : http://compass.xbox.com/assets/ff/eb/ffebf3ba-cc7c-44d3-a801-2a8e8cb16c9e.png?n=Kinect-Sensor_blade-console-kinect_576x350.png

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2.1 ส่วนประกอบและคุณสมบัติของ Kinect ONE



รูปที่ 2.6 ส่วนประกอบของ Kinect ONE

ที่มา : https://image.slidesharecdn.com/nuiebiometricsinwindows10-151019081222-lva1-app6891/95/nui-e-biometrics-in-windows-10-8-638.jpg?cb=1445242569

1) Infrared Optics (3D Depth Sensors)

ค่าขอบเขตการมองเห็น (Field Of View) แนวตั้ง 60 องศา แนวนอน 70 องศา ใช้ในการตรวจจับตำแหน่งของผู้เล่น โดยสามารถวิเคราะห์ได้ทั้ง 3 แกน ได้แก่ กว้าง ยาว ระยะใกล้ - ไกล และตรวจจับได้มากถึง 25 ข้อต่อ (รวมนิ้วมือ) อีกทั้งยังสามารถใช้งานได้พร้อมกันถึง 6 คน ประกอบด้วย 2 ส่วน ได้แก่ *Infrared Laser Projector* ทำหน้าที่จะส่งแสงอินฟราเรดไปกระทบกับวัตถุที่อยู่ในแนวรัศมีและ *Monochrome CMOS Sensor* ทำหน้าที่เป็นตัวรับ หลังจากแสงอินฟราเรดตกกระทบกับวัตถุแล้ว

2) RGB Camera 1280 x 960 pixels

กล้องที่เก็บข้อมูลด้วยความละเอียด 1280x960 ทำให้สามารถจับภาพสีได้รวมทั้งยังสามารถถ่ายภาพเคลื่อนไหวได้ ซึ่งช่วยในระบบจดจำใบหน้า (Face recognition) และการตรวจจับลักษณะเด่นอื่นๆ โดยใช้ความสามารถในการตรวจจับองค์ประกอบของสีทั้งสามสี ได้แก่ สีแดง สีเขียว และสีน้ำเงิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) Multi-Array Microphone จำนวน 4 ตัว

ทำหน้าที่ใช้รับเสียงจากผู้เล่น สามารถบันทึกและแยกแยะเสียงของผู้เล่น และเสียงรบกวนที่อยู่บริเวณรอบๆ ได้ รวมทั้งสามารถระบุตำแหน่งของแหล่งกำเนิดเสียงและทิศทางคลื่นเสียงของผู้ใช้ได้

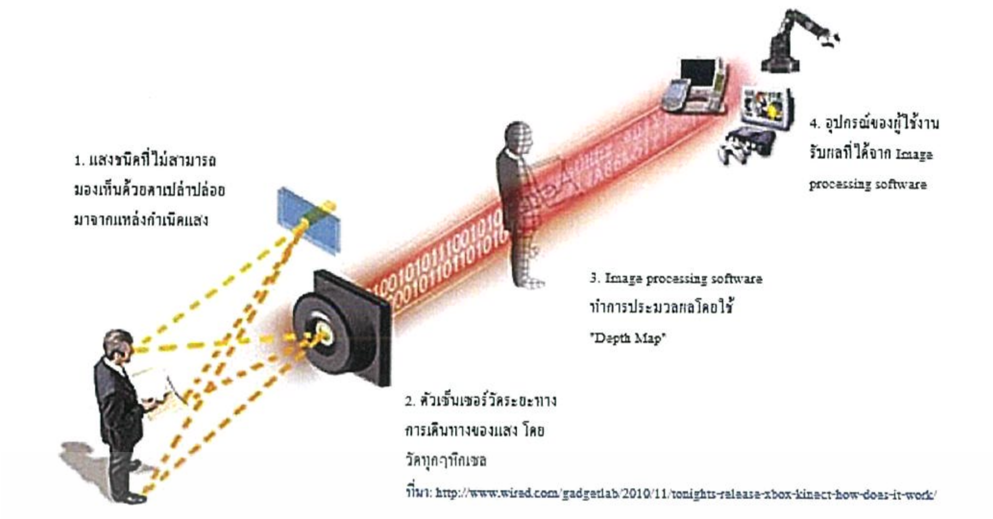
ตารางที่ 2.2 คุณสมบัติของ Kinect ONE

Property	Spec
Field of View (Horizontal, Vertical, Diagonal)	70° H, 60° V
Color Camera	1920 x 1080 @30 fps
Depth Camera	512 x 424
Maximum Depth Distance	4.5 M
Minimum Depth Distance	50 cm
Skeleton Joints Defined	26 joints
Full Skeletons Tracked	6
Audio: built-in microphones	4 microphones
Data interface	USB 3.0
Power supply	USB 3.0

2.1.3 หลักการทำงานของ Kinect

ภายใน Kinect ประกอบด้วย อุปกรณ์ฉายแสงอินฟราเรด กล้องวัดความลึกของภาพ กล้องวีดีโอ ไมโครโฟน และเซนเซอร์ประมวลผล มีการทำงานเริ่มจากการฉายแสงอินฟราเรดออกจากตัว Kinect ซึ่งไม่สามารถมองเห็นได้ด้วยตาเปล่า แสงที่ถูกฉายออกมาจะมีลักษณะเป็นจุดๆ ตามแนวตั้งและแนวนอน หลังจากนั้นกล้องวัดความลึก ในส่วน Infrared Laser Projector จะส่งแสงอินฟราเรดไปกระทบกับวัตถุที่อยู่ในแนวรัศมี เมื่อแสงอินฟราเรดกระทบกับวัตถุก็จะสะท้อนกลับมาที่ตัวรับ Monochrome CMOS Sensor จากนั้นซอฟต์แวร์ก็จะคำนวณระยะทาง โดยจะวัดจากระดับความสว่างของแสงอินฟราเรดที่ตกกระทบลงบนวัตถุ หากความสว่างมีมากแสดงว่าวัตถุนั้นอยู่ใกล้ จากนั้นจะส่งไปให้เซนเซอร์เพื่อทำการวัดความลึกตามแนวแกน Z ทำให้สามารถจำลองสภาพแวดล้อมเป็นสามมิติได้ นอกจากนี้ Kinect ยังสามารถตรวจจับการเคลื่อนไหว 3 มิติแบบเต็มตัว จดจำเสียง และจดจำใบหน้าของผู้เล่น รวมทั้งสามารถใช้เสียงในการควบคุมการใช้งานได้อีกด้วย (Voice Controller)

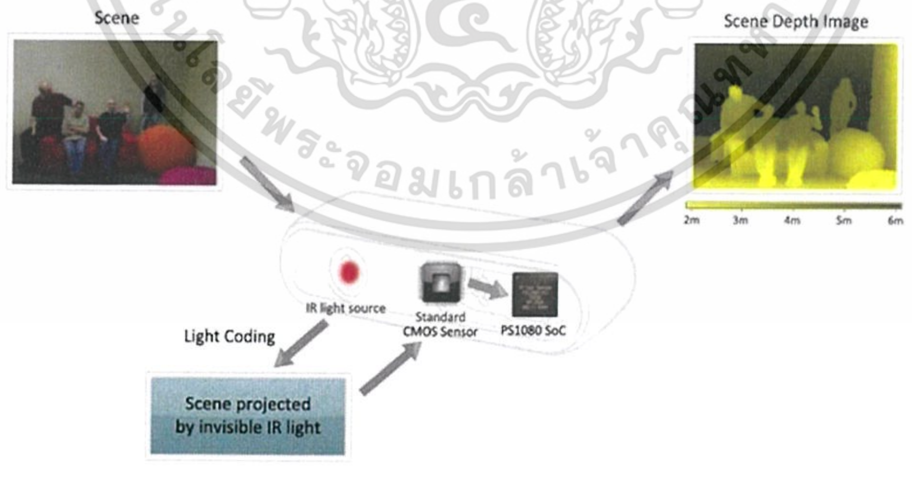
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 แสดงการทำงานของระบบตรวจวัดระยะ

ที่มา : https://kinectasia.files.wordpress.com/2012/03/canesta-howitworks1.jpg

เมื่อได้ระดับความลึกของภาพแล้ว ทำให้เซนเซอร์ของ Kinect สามารถแยกผู้เล่นออกจากสภาพแวดล้อมภายในห้องได้เช่น ผนัง เฟอร์นิเจอร์ต่างๆ หรือแม้แต่การจำแนกว่ามือของผู้เล่นอยู่ข้างหน้าหรือข้างหลัง ซึ่งมีประโยชน์มากในเกมประเภทที่ต้องใช้การแสดงท่าทางการตี เช่น ในเกมตีปิงปอง ใช้ในการวัดระดับความแรงของการตีลูกปิงปอง ส่วนกล้องวีดีโออีกหนึ่งชิ้นนั้น ใช้สำหรับรับภาพผู้เล่นเข้าไปแสดงในเกม หรือหากเป็นเกมประเภทเต้น ที่ผู้เล่นสามารถเลือกได้ว่า จะให้แสดงตัวละครในเกมเป็นตัวผู้เล่นเอง เป็นต้น



รูปที่ 2.8 การฉายแสงอินฟราเรดเพื่อระบุความลึกของวัตถุ

ที่มา : https://d3nevzfk7ii3be.cloudfront.net/igi/TG5BJRKoMQdJoLLA.medium

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4 การใช้งาน Kinect สำหรับ Windows

Software Development Kit (SDK) ประกอบด้วยไดรฟ์เวอร์ต่างๆของ Kinect สำหรับคอมพิวเตอร์พีซีรวมอยู่ด้วย ซึ่งโปรแกรมที่สามารถนำมาใช้พัฒนาแอพพลิเคชั่น Kinect ได้แก่ C++ C# และ Visual Basic ส่วนคุณสมบัติต่างๆที่ถูกรวมไว้ใน SDK ด้วย ได้แก่

1) Raw Sensor Streams สำหรับนำข้อมูลดิบที่ได้จากเซนเซอร์ต่างๆ ไปใช้งาน
2) Skeletal Tracking สำหรับติดตามโครงร่างของมนุษย์ ซึ่งสามารถติดตามได้สูงสุด 6 คนพร้อมกัน ความโดดเด่นของคุณสมบัตินี้คือ สามารถนำไปพัฒนาเป็นแอพพลิเคชั่นที่ควบคุมด้วยท่าทาง (Gestures Control)
3) Advanced Audio Capabilities ใช้ในการจำแนกเสียงต่างๆ ซึ่งสามารถกำจัดเสียงรบกวนต่างๆ (Noise Suppression and Echo Cancellation) และแยกเสียงของแต่ละบุคคลได้ ซึ่งคุณสมบัตินี้สามารถไปนำไปพัฒนาเป็นการควบคุมด้วยเสียง (Voice Recognition)
4) โค้ดโปรแกรมตัวอย่างและเอกสารประกอบ

Kinect Sensor จัดเป็นอินเทอร์เฟสชนิดไม่ต้องสัมผัส (Touch-less UI) หรือเป็นอินเทอร์เฟสประเภท Natural UI (NUI) คือ ใช้การเคลื่อนไหวของส่วนต่างๆของร่างกายมนุษย์ โดย Kinect นั้นจะติดต่อกับแอพพลิเคชั่นผ่านทาง NUI Library ซึ่งข้อมูลที่ Kinect ป้อนให้แก่ NUI Library มีสามชนิดด้วยกัน คือ

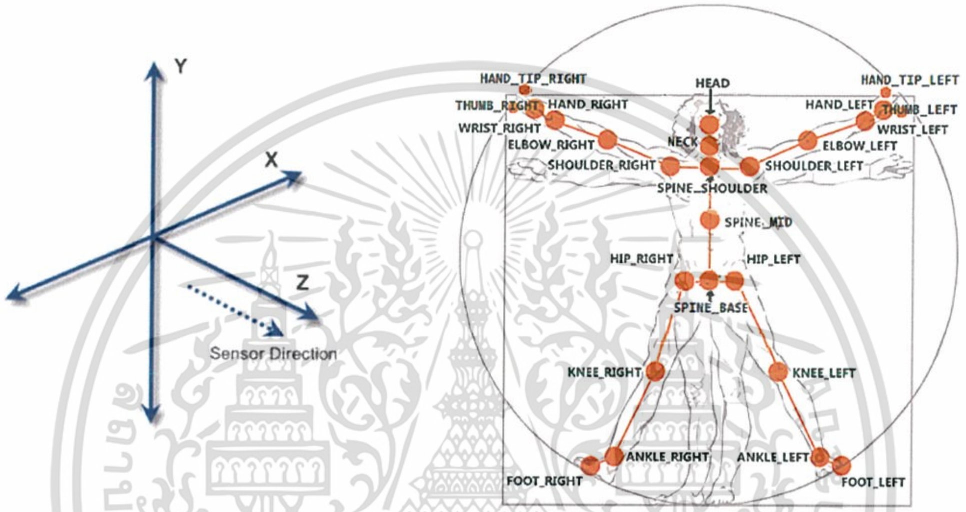
1) Image Stream (เซ็นเซอร์ที่จับภาพคือกล้อง VGA และ Monochrome)
2) Depth Stream (เซ็นเซอร์ที่ตรวจจับความตื้นลึกคือกล้องอินฟราเรด)
3) Audio Stream (เซ็นเซอร์ที่รับสัญญาณเสียงคือไมโครโฟนแบบอาเรย์)

ส่วนสำคัญของ Kinect นั้นคือ NUI API ซึ่งเป็นส่วนเชื่อมต่อระหว่าง Kinect Sensors (VGA Camera, Monochrome Camera, IR Camera, and 4 Microphones) กับคอมพิวเตอร์ นอกจากนี้หน้าที่สำคัญอีกอย่างหนึ่งคือการนำข้อมูลต่างๆ เช่น รูปภาพ ความตื้นลึกของรูป ไปใช้ต่อ โดยข้อมูลเหล่านี้เมื่อถูกประมวลผลแล้วสามารถนำไปใช้งานในรูปแบบ Skeletal Tracking

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5 การตรวจจับการเคลื่อนไหวโดยใช้ Kinect

Kinect มีระบบการรับรู้การเคลื่อนไหวของผู้เล่นอยู่ภายในเซนเซอร์เอง ซึ่งเป็นการนำเทคโนโลยีปัญญาประดิษฐ์เข้ามาช่วย โดยจะมีการส่งข้อมูลการเคลื่อนไหวของผู้เล่นในลักษณะต่างๆเข้าไปเป็นจำนวนมาก ไม่ว่าจะเป็นท่าทางการนั่ง ยืน การเอียงตัว การขว้างลูกบอล การหยิบสิ่งของ หรือแม้แต่กระทั่งการตวัดมือ ข้อมูลการเคลื่อนไหวเหล่านี้จะถูกประมวลเหลือเพียงการเคลื่อนไหวของโครงกระดูกซึ่ง Kinect จะวิเคราะห์ลักษณะการเคลื่อนไหวของข้อต่อแต่ละข้อต่อ เพื่อนำไปวิเคราะห์อีกครั้งว่าขณะนี้ผู้เล่นกำลังแสดงท่าทางอะไรอยู่

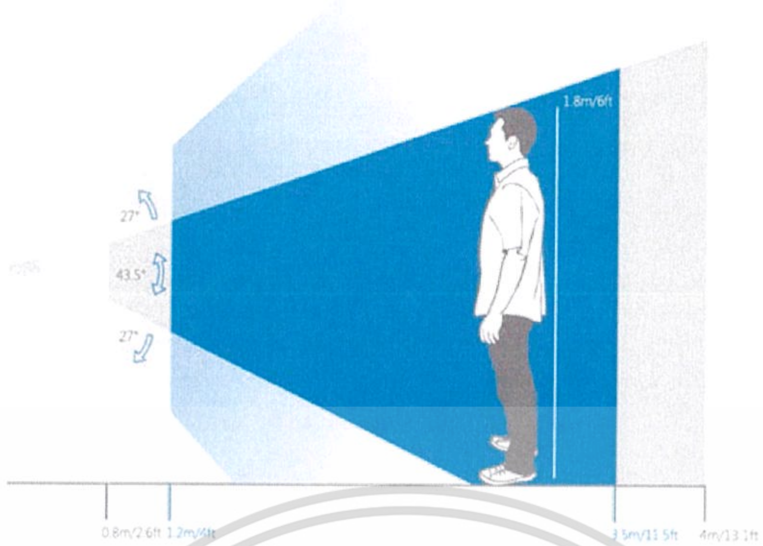


รูปที่ 2.9 ตำแหน่งของข้อต่อที่ใช้ในการวิเคราะห์

ที่มา : https://i-msdn.sec.s-msft.com/dynimg/IC741371.png

ตำแหน่งของจุดกำเนิด (X=Y=Z = 0) จะเป็นตำแหน่งของกล้องที่ใช้ในการตรวจจับ แกน Z จะเป็นทิศทางที่กล้องตรวจจับ ค่า X จะเป็นค่าแสดงระยะทางที่ห่างออกไปจากจุดกำเนิดตามแนวนอนสัมพัทธ์กับกล้องที่ตรวจจับ จะมีค่าเป็นบวกเมื่อตำแหน่งที่ถูกตรวจจับได้อยู่ทางด้านขวาของกล้อง ค่า Y จะเป็นค่าแสดงระยะทางที่ห่างออกไปจากจุดกำเนิดตามแนวตั้งที่กล้องตรวจจับ จะมีค่าเป็นบวกเมื่อตำแหน่งที่ถูกตรวจจับได้อยู่สูงกว่าตำแหน่งของกล้อง และค่า Z จะเป็นระยะทางที่ห่างออกไปจากกล้องโดยจะมีค่าเป็นบวกเสมอ โดยระยะการตรวจจับการเคลื่อนไหวของ Kinect จะสามารถตรวจจับได้ใกล้ที่สุดในระยะที่ห่างออกไป 0.8 เมตร จนถึง 4 เมตร แต่ระยะที่สามารถตรวจจับได้ดีที่สุดคือ ระยะห่างจาก Kinect ตั้งแต่ 1.2 เมตร จนถึง 3.5 เมตร (ระยะห่างอุดมคติที่ Microsoft แนะนำคือ ประมาณ 2.5 เมตร ซึ่งค่าที่ได้จากการตรวจจับจะใกล้เคียงกับการวัดจริงมากที่สุด)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 ระยะห่างและมุมที่กล้องสามารถจับได้
ที่มา : https://mcu56.learninginventions.org/wp-content/uploads/2014/01/IC5843941.png

เมื่อผู้ที่ถูกตรวจจับยืนหันหน้าเข้าหากล้อง ตำแหน่งของข้อต่อข้างซ้ายและขวาจะสลับข้างกันแตกต่างจากร่างกายจริง แต่ละตำแหน่งของข้อต่อจะมีค่า X, Y และ Z ในหน่วยมิลลิเมตร และมีฟิลด์ Confidence แสดงสถานะการตรวจจับตำแหน่งข้อต่อนั้นๆ โดยจะมีค่าระหว่าง 0 – 1 ซึ่ง 1 หมายถึง สามารถตรวจจับได้ และหากน้อยกว่า 1 หมายถึง การตรวจจับเป็นการประมาณ

ในการตรวจจับการเคลื่อนไหว หากผู้ที่ถูกตรวจจับอยู่ภายในมุมกล้อง ห่างจากกล้องประมาณ 1.2 – 3.5 เมตร ค่าตำแหน่งของข้อต่อที่ถูกตรวจจับได้ จะมีค่าใกล้เคียงกับการวัดจริง และฟิลด์ Confidence จะมีค่าเป็น 1 แต่หากมีบางข้อต่อที่กล้องไม่สามารถตรวจจับได้หรือถูกบดบัง เช่น ยืนมือไพล่หลัง เป็นต้น ค่าตำแหน่งของข้อต่อนั้นจะเป็นค่าประมาณและฟิลด์ Confidence จะมีค่าน้อยกว่า 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.6 คำแนะนำเพิ่มเติมจาก Microsoft ในการตรวจจับการเคลื่อนไหวด้วย Kinect

1) ผู้ใช้ไม่ควรสวมชุดที่หลวมโคร่ง ในขณะที่ทำการตรวจจับการเคลื่อนไหว เพราะจะทำให้ไม่สามารถตรวจจับตำแหน่งของข้อต่อได้

2) การตรวจจับข้อต่อที่แขนจะมีความเสถียรน้อย หากแขนนั้นใกล้ชิดอยู่กับส่วนของร่างกาย โดยเฉพาะอยู่ชิดกับลำตัว ถ้าแขนทั้งสองอยู่ชิดกับลำตัวหรืออยู่ใกล้กับส่วนอื่นอาจรวมแขนนั้นเป็นส่วนหนึ่งของร่างกายส่วนนั้นได้ และจะทำให้ไม่สามารถตรวจจับข้อต่อที่แขนนั้นได้

3) การตรวจจับข้อต่อที่ขาจะมีการตรวจจับที่ดีขึ้น หากผู้ใช้ไม่ยืนชิดขา

4) การเคลื่อนไหวที่รวดเร็วหรือการแสดงท่าทางที่ซับซ้อน อาจทำให้ไม่สามารถตรวจจับการเคลื่อนไหวได้ เช่นการทำท่าเตะฟุตบอลอย่างรวดเร็ว เป็นต้น

จากการใช้งานกล้อง Kinect ในการตรวจจับการเคลื่อนไหวของร่างกาย ผู้จัดทำพบว่าตำแหน่งที่ดีที่สุดในการตั้งกล้องตรวจจับการเคลื่อนไหวคือ การตั้งกล้องให้เผชิญหน้ากับผู้ใช้ และไม่มีสิ่งที่บดบังการตรวจจับ เพราะส่วนของร่างกายเกือบทุกๆ ส่วนจะสามารถตรวจจับได้ตลอดเวลา และหากมีสิ่งของที่อยู่ใกล้กับผู้ใช้ เช่น ผู้ใช้นั่งอยู่บนเก้าอี้ เป็นต้น กล้องอาจรวมสิ่งของที่อยู่ใกล้นั้นเป็นส่วนหนึ่งของร่างกายผู้ใช้และระบุตำแหน่งของข้อต่อที่ผิดพลาดได้ การระบุส่วนของร่างกายด้านซ้ายและขวาอาจสามารถสลับกันได้เนื่องจาก Kinect ไม่สามารถวิเคราะห์ภาพโครงร่างมนุษย์ที่กล้องตรวจจับได้ว่าเป็นด้านหน้าหรือด้านหลังของร่างกาย หากผู้ใช้หันหลังให้กับกล้องตั้งแต่เริ่มการตรวจจับการเคลื่อนไหว ตำแหน่งของข้อต่อที่ควรจะเป็นด้านซ้ายก็จะสลับไปเป็นด้านขวาได้ นอกจากนี้ Kinect ยังสามารถตรวจจับและให้ค่าตำแหน่งสิ่งของที่มีรูปร่างคล้ายกับโครงร่างมนุษย์ได้ เช่น พัดลมตั้งพื้น เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 Visual Studio 2013

Visual Studio (วิชวล-สตูดิโอ) เป็นโปรแกรมหรือเครื่องมือที่ช่วยในการเขียนโปรแกรมคอมพิวเตอร์ด้วยภาษา C# ซึ่งโปรแกรมนี้ได้พัฒนาขึ้นครั้งแรกโดยบริษัทไมโครซอฟท์ในชื่อของ Visual C# .NET (วิชวล-ซี-ชาร์ป-ดอตเน็ต)

โปรแกรมนี้ได้รับความนิยมอย่างสูงในเวอร์ชันถัดมา คือ Visual C# 2005 โดยมาพร้อมกับ Visual Studio 2005 และภายหลังได้มีการรวมความสามารถในการพัฒนาโปรแกรมของ Visual C# ให้อยู่ในชุดโปรแกรมเดียวกันใน Visual Studio 2013

โดย Visual Studio จัดเป็นซอฟต์แวร์ประเภท IDE (Integrated Development Environment) ซึ่งเป็นการนำแนวความคิดการทำงานแบบรวมศูนย์มาใช้ คือ การทำให้วงจรการพัฒนาระบบทั้งหมดทำงานได้อย่างสะดวก รวดเร็ว และ ง่ายดาย เริ่มตั้งแต่การวิเคราะห์ ออกแบบ จนถึงการนำไปปรับใช้ให้เหมาะสมกับวัตถุประสงค์ของการเขียนโปรแกรมคอมพิวเตอร์นั้นๆ

2.2.1 ภาษา C#

ภาษา C# (ซี-ชาร์ป) เป็นภาษาโปรแกรมคอมพิวเตอร์ระดับสูงที่ใช้สำหรับเขียนโปรแกรมคอมพิวเตอร์ ที่ได้รับความนิยมเพิ่มมากขึ้นเรื่อยๆ ในปัจจุบัน และเป็นภาษาโปรแกรมคอมพิวเตอร์ที่เหมาะสาหรับผู้ที่เริ่มต้นสนใจที่จะเขียนโปรแกรมคอมพิวเตอร์เป็นอย่างยิ่ง ซึ่งภาษา C# ถูกพัฒนามาจากภาษา C++ และมีโครงสร้างแบบเชิงวัตถุ (object-oriented programming) โดยใช้โปรแกรม Visual Studio เป็นเครื่องมือสำหรับพัฒนาโปรแกรมคอมพิวเตอร์ ทำให้ผู้เขียนโปรแกรมสามารถพัฒนาโปรแกรมคอมพิวเตอร์ได้ไม่ยากนัก

2.2.2 ข้อดีของภาษา C#

1) เป็นภาษาที่เขียนง่าย ไม่ซับซ้อนและเรียบง่าย เพราะคล้ายภาษา Java ภาษา C และ ภาษา C++ ทำให้หลายคนเข้าใจได้ไม่ยาก

2) เป็นภาษาโปรแกรมคอมพิวเตอร์ยุคใหม่ที่ถูกสร้างขึ้นมาสำหรับการพัฒนาโปรแกรมคอมพิวเตอร์ ภายใต้แนวคิด .NET Framework ซึ่งเป็นแนวคิดที่ได้รับความนิยมสูงที่สุดในปัจจุบัน

3) เป็นภาษาที่ถูกออกแบบมาให้ทำงานบน .NET Framework (ดอตเน็ต-เฟรมเวิร์ก) ซึ่งมีคุณสมบัติที่สำคัญคือ ผู้ใช้งานสามารถใช้งานบนระบบฮาร์ดแวร์ (Hardware) หรือระบบปฏิบัติการ (Operating System) ที่แตกต่างกันได้อย่างไม่มีปัญหา เช่น เครื่องพีซีกับเครื่องแมคหรือ ระบบปฏิบัติการวินโดว์กับระบบปฏิบัติการแมคอินทอช เป็นต้น ดังนั้นผู้เขียนโปรแกรมจึงสามารถเขียนโปรแกรมคอมพิวเตอร์ใหม่ๆ ได้โดยง่าย รวดเร็ว และไม่ต้องติดข้อจำกัดต่างๆ อย่างเช่นการเขียนโปรแกรมคอมพิวเตอร์ในสมัยก่อนอีกต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) เป็นภาษาที่แข็งแกร่ง เพราะเป็นภาษาที่ได้มีการแก้ไขข้อบกพร่องบางอย่างของภาษา Java ภาษา C และ ภาษา C++ เหล่านั้น ทำให้ ภาษา C# เป็นภาษาที่มีความสมบูรณ์ตามแบบฉบับของโครงสร้างแบบเชิงวัตถุ (object-oriented programming)

2.2.3 ส่วนต่างๆ ของโปรแกรม Visual Studio

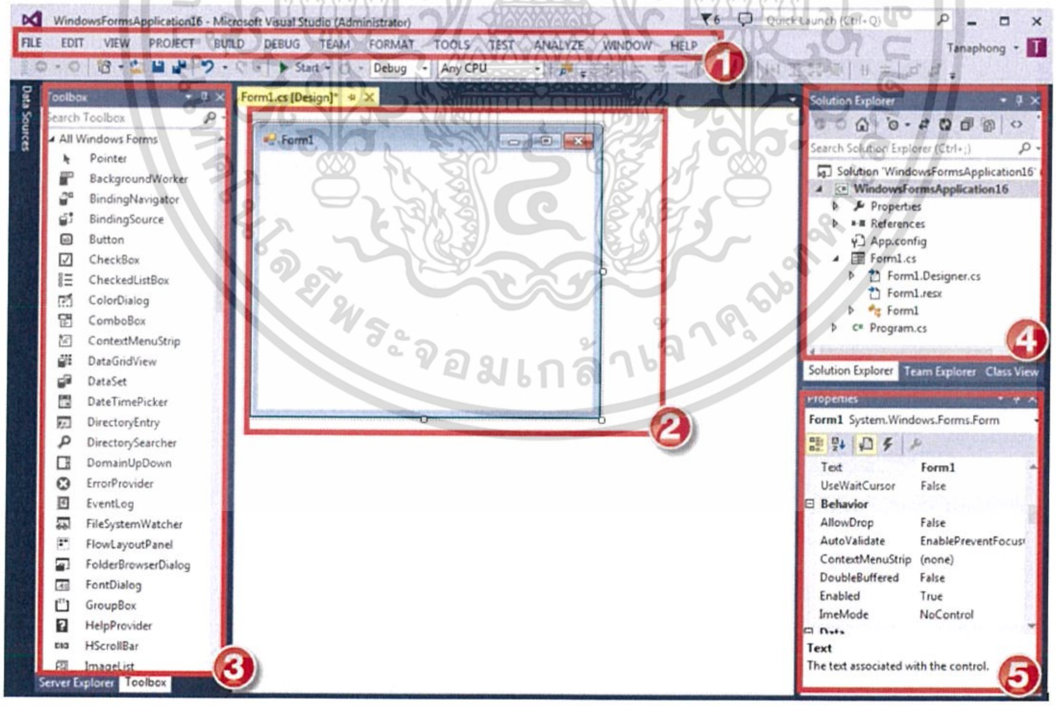
1) แถบเมนู (Menu bar) เป็นแถบแสดงรายการคำสั่งต่างๆ ของโปรแกรม เช่น การสร้างโปรเจคใหม่ การตั้งค่าโปรแกรม หรือการเรียกใช้เครื่องมืออื่นๆ เป็นต้น

2) พื้นที่สำหรับเขียนโปรแกรม (Workspace) เป็นพื้นที่สำหรับการออกแบบและเขียนคำสั่งต่างๆ ให้โปรแกรม

3) กล่องเครื่องมือ (Toolbox) เป็นกลุ่มของเครื่องมือต่างๆ สำหรับใช้ในการออกแบบหน้าจอของโปรแกรม เช่น Button, Listbox, Checkbox, Radio button Label เป็นต้น

4) ส่วนควบคุมการทำงาน (Solution Explorer) เป็นส่วนสำหรับแสดงโปรเจ็กหรือไฟล์ที่กำลังเปิดอยู่

5) หน้าต่างแสดงคุณสมบัติของเครื่องมือ (Properties) เป็นส่วนที่กำหนดคุณสมบัติต่างๆ ให้กับเครื่องมือแต่ละตัว เช่น การกำหนดสี กำหนดขนาดตัวอักษร และอื่นๆ



รูปที่ 2.11 ส่วนต่างๆ ของโปรแกรม Visual Studio
ที่มา : http://tanateach.com/wordpress/topic/2-1/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 เครื่องมือพื้นฐานที่นำมาประยุกต์ใช้งาน

1) เครื่องมือ Textbox

Textbox หรือ กล่องข้อความ เป็นเครื่องมือสำหรับใช้ในการรับค่า (Input) จากผู้ใช้ผ่านทางคีย์บอร์ด หรือแสดงค่า (Output) ออกมาแสดงผล โดยข้อมูลที่รับ/แสดง ด้วย Textbox นั้นจะเป็นข้อมูลชนิดตัวอักษร หรือ String สามารถใช้เครื่องมือ Textbox สามารถเรียกได้จากแถบกล่องเครื่องมือ (Toolbox)

2) เครื่องมือ Button

Button หรือ ปุ่ม เป็นเครื่องมือสำหรับใช้ในการสั่งให้โปรแกรมทำงานตามคำสั่งใดๆ เมื่อผู้ใช้กดที่ปุ่มนั้นๆ (Event Click) ซึ่งสามารถเรียกใช้ Button ได้จากแถบกล่องเครื่องมือ (Toolbox) เครื่องมือ Button เป็นเพียงเครื่องมือที่คอยรับคำสั่งโดยการคลิกจากผู้ใช้เท่านั้น ดังนั้นเราสามารถเขียนคำสั่งใดๆ เพื่อให้โปรแกรมทำงานได้ ภายใต้ Event Click ของ Button เช่น กำหนดให้ เมื่อผู้ใช้กดปุ่ม Button ค่าต่างๆ ที่แสดงจะถูกบันทึกลงในฐานข้อมูล

3) เครื่องมือ Label

Label หรือ ป้ายข้อความ เป็นเครื่องมือที่ใช้สำหรับการแสดงข้อความโดยเฉพาะ ไม่สามารถรับค่าจากผู้ใช้ได้ โดยข้อมูลที่จะแสดงผ่าน label นั้นจะต้องเป็นข้อมูลประเภท String เช่นเดียวกับ Textbox ซึ่งเราสามารถเรียกใช้ Label ได้จากแถบกล่องเครื่องมือ (Toolbox) และสามารถนำไปวางที่ตำแหน่งต่างๆ บนฟอร์มของโปรแกรมได้ตามต้องการ

2.2.5 ระบบปฏิบัติการที่ใช้งานได้

1) Windows 8
2) Windows 7 SP1
3) Windows Server 2012 R2
4) Windows Server 2012
5) Windows Server 2008 R2 SP1

2.2.6 ฮาร์ดแวร์ขั้นต่ำที่สามารถใช้งานร่วมกับโปรแกรม Visual Studio 2013 ได้

1) ซีพียูความเร็ว 1.6 GHz ขึ้นไป
2) แรมไม่ต่ำกว่า 1 GB
3) พื้นที่ในฮาร์ดดิสก์ไม่ต่ำกว่า 20 GB
4) หน้าจอแสดงผลความละเอียดไม่ต่ำกว่า 1024 x 768 พิกเซล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 XAMPP

รูปที่ 2.12 สัญลักษณ์ของโปรแกรม XAMPP
ที่มา : http://anisadesign.esy.es/Xampp.html

XAMPP พัฒนาโดยโครงการ Apache Friends ที่เป็นโครงการไม่แสวงหาผลกำไร ที่จัดตั้งในปี ค.ศ. 2002 โดย Kai 'Oswald' Seidler และ Kay Vogelgesang ทั้งนี้ XAMPP ประกอบด้วยโปรแกรมย่อยได้แก่โปรแกรม Apache โปรแกรมฐานข้อมูล MySQL โปรแกรมภาษา PHP และภาษา Perl XAMPP มีการปรับปรุงรุ่นอย่างต่อเนื่องเพื่อให้ทำงานได้กับระบบปฏิบัติการทั้ง Microsoft Windows, Mac OS x, Linux, Solaris และไม่มีค่าใช้จ่ายในการดาวน์โหลดใช้งาน

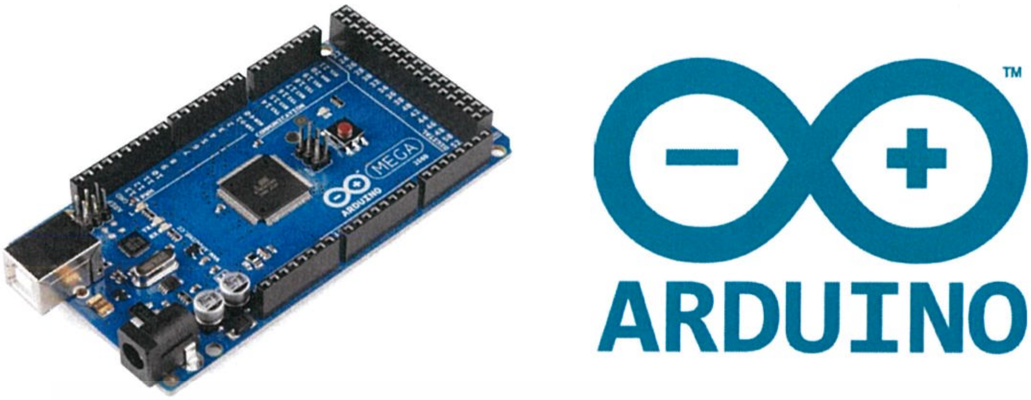
XAMPP เป็นโปรแกรม Apache web server ไว้จำลอง web server เพื่อไว้ทดสอบสคริปหรือเว็บไซต์ในเครื่องของเรา โดยที่ไม่ต้องเชื่อมต่ออินเตอร์เน็ตและไม่ต้องมีค่าใช้จ่ายใดๆ ง่ายต่อการติดตั้งและใช้งานโปรแกรม Xampp จะมาพร้อมกับ PHP ภาษาสำหรับพัฒนาเว็บแอพพลิเคชั่นที่เป็นที่นิยม, MySQL ฐานข้อมูล, Apache ทำหน้าที่เป็นเว็บเซิร์ฟเวอร์, Perl อีกทั้งยังมาพร้อมกับ OpenSSL, phpMyadmin (ระบบบริหารฐานข้อมูลที่พัฒนาโดย PHP เพื่อใช้เชื่อมต่อไปยังฐานข้อมูล สนับสนุนฐานข้อมูล MySQL และ SQLite โปรแกรม Xampp จะอยู่ในรูปแบบของไฟล์ Zip, tar, 7z หรือ exe

2.3.1 สิ่งที่รวมมาใน XAMPP

1) Apache 2.4.23
2) MariaDB 10.1.16 (MySQL)
3) PHP 5.5.38
4) phpMyAdmin 4.5.1
5) OpenSSL 1.0.2
6) XAMPP Control Panel 3.2.2
7) Webalizer 2.23-04
8) Mercury Mail Transport System 4.63
9) FileZilla FTP Server 0.9.41
10) Tomcat 7.0.56 (with mod_proxy_ajp as connector)
11) Strawberry Perl 7.0.56 Portable

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 Arduino



รูปที่ 2.13 บอร์ด Arduino และเครื่องหมายทางการค้า
ที่มา : https://www.gravitechthai.com/upload/guru/pictures/pic-3599964342.png

Arduino อ่านว่า (อา-ดู-อิ-โน่ หรือ อาดุยโน่) เป็นบอร์ดไมโครคอนโทรเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software บอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ด หรือโปรแกรมต่อได้อีกด้วย

ความง่ายของบอร์ด Arduino ในการต่ออุปกรณ์เสริมต่างๆ คือผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิคส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด หรือเลือกต่อกับบอร์ดเสริม (Arduino Shield) ประเภทต่างๆ เช่น Arduino XBee Shield, Arduino Music Shield, Arduino Relay Shield, Arduino Wireless Shield, Arduino GPRS Shield เป็นต้น โดยนำมาเสียบเข้ากับบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้เลย

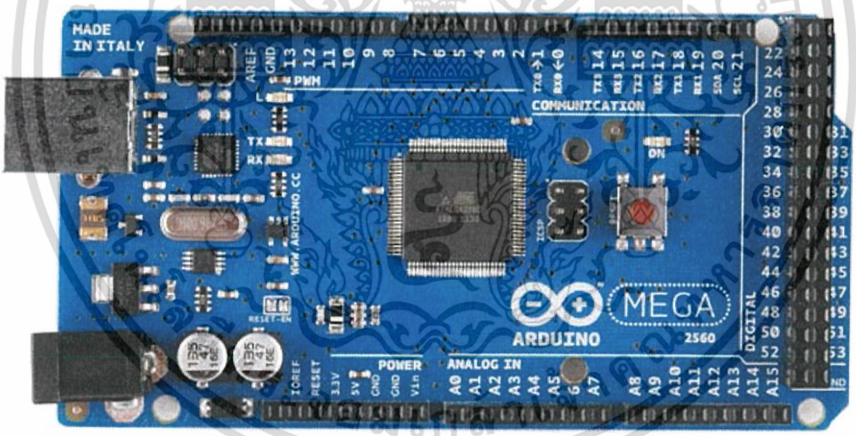
2.4.1 Arduino MEGA 2560

Arduino Mega 2560 บอร์ดรุ่นใหญ่ในของตระกูล Arduino มีคุณสมบัติต่างๆ เพิ่มขึ้นจาก Arduino Uno R3 ประกอบด้วยชิพ ATmega2560 ที่มีหน่วยความจำแฟลช 256 KB แรม 8 KB ใช้ไฟเลี้ยง 7 ถึง 12 V แรงดันของระบบอยู่ที่ 5 V มี Digital Input / Output มากถึง 54 ขา (เป็น PWM ได้ 14 ขา) มี Analog Input 16 ขา Serial UART 4 ชุด I2C 1 ชุด SPI 1 ชุด เขียนโปรแกรมบน Arduino IDE และโปรแกรมผ่าน USB เหมาะสำหรับผู้ที่สนใจเริ่มต้นเรียนรู้การพัฒนาไมโครคอนโทรลเลอร์ที่ต้องการบอร์ด Arduino ที่มีหน่วยความจำและขาสัญญาณต่างๆ ให้ต่อใช้งานมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 คุณสมบัติของ Arduino 2560

Property	Spec
Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output, 4 UART TTL)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz



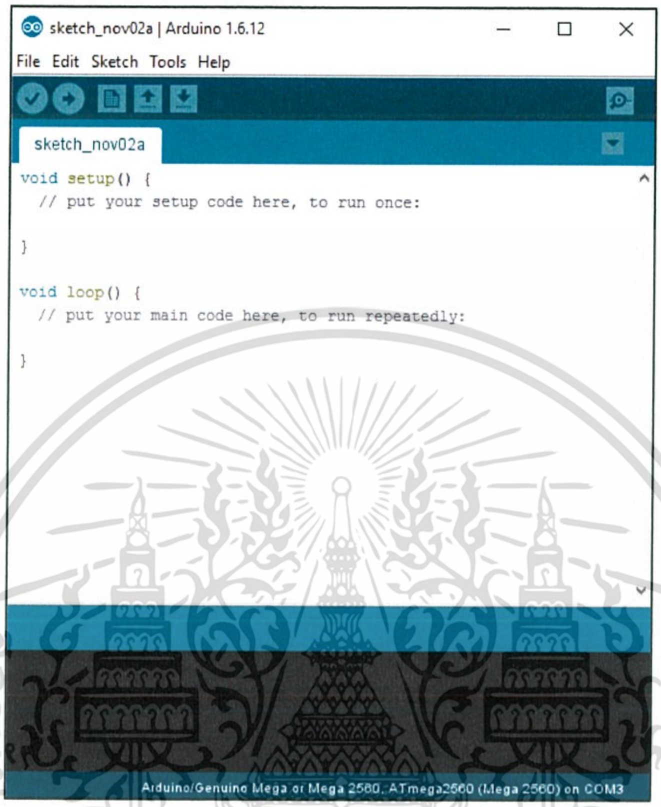
รูปที่ 2.14 Arduino MEGA 2560

ที่มา : http://www.myarduino.net/product/32/arduino-mega-2560-r3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

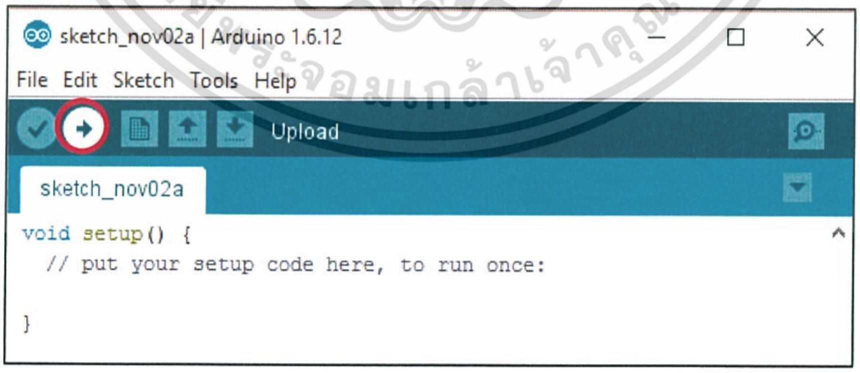
2.4.2 Arduino Integrated development environment (IDE)

Arduino IDE คือ ซอฟต์แวร์ที่ใช้สำหรับเขียนโปรแกรม เพื่อทำงานร่วมกับบอร์ด Arduino ทุกรุ่น โดยสามารถเขียนโปรแกรมด้วยภาษา C++ และทำการเชื่อมต่อกับส่วนฮาร์ดแวร์ (บอร์ด Arduino) อย่างสะดวกและรวดเร็วผ่านสาย USB



รูปที่ 2.15 หน้าต่างหลักของโปรแกรม Arduino IDE

ที่มา : ผู้จัดทำ



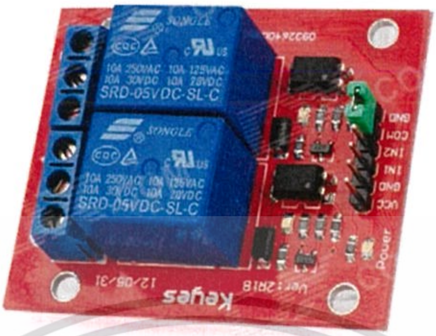
รูปที่ 2.16 ปุ่ม Upload ใช้สำหรับอัพเดทโปรแกรมไปยังบอร์ด Arduino

ที่มา : ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 2 Channels Keyes Relay

บอร์ดรีเลย์ขนาด 2 ช่อง มีเอาต์พุตคอนเน็คเตอร์ที่รีเลย์เป็น NO/COM/NC สามารถใช้กับโหลดได้ทั้งแรงดันไฟฟ้า DC และ AC โดยใช้สัญญาณในการควบคุมการทำงานด้วยสัญญาณโลจิก TTL



รูปที่ 2.17 Keyes Relay 2R1B

ที่มา : https://s-media-cache-ak0.pinimg.com/originals/48/26/fd/4826fda01c25a013a3d41fb55d010d1d.jpg

2.5.1 คุณสมบัติ (Features)

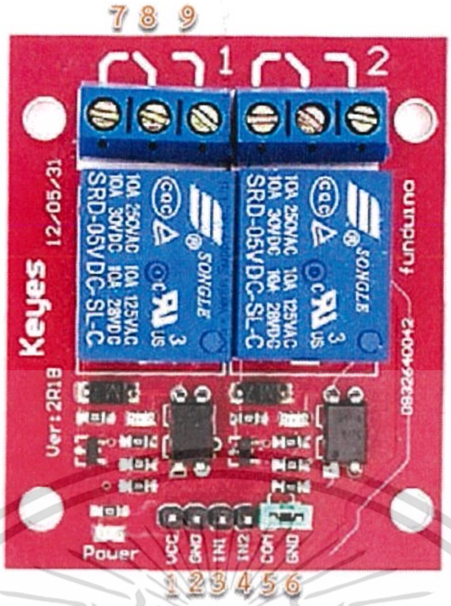
1) รีเลย์เอาต์พุตแบบ SPDT จำนวน 2 ช่อง
2) สั่งงานด้วยระดับแรงดัน TTL
3) CONTACT OUTPUT รับแรงดันได้สูงสุด 250 VAC 10 A , 30 VDC 10 A
4) มี LED แสดงสถานะ การทำงานของรีเลย์และแสดงสถานะของบอร์ด
5) มีจัมพ์เปอร์สำหรับเลือกว่าจะใช้กราวด์ร่วมหรือแยก
6) มี OPTO-ISOLATED เพื่อแยกกราวด์ส่วนของสัญญาณควบคุมกับไฟที่ขับรีเลย์ออกจากกัน

2.5.2 คุณลักษณะ (Specification)

1) ควบคุมไฟ DC ได้สูงสุด 30VDC 10A และ ไฟ AC สูงสุด 250VAC 10A
2) ระดับสัญญาณอินพุตควบคุมแบบ TTL ทำงานด้วยสัญญาณแบบ Active High
3) ขนาดรูยึดบอร์ด 3mm
4) ขนาด (L x W x H): 55 x40 x 20 mm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3 ขาสัญญาณและการเชื่อมต่อ (Pin Definition and Rating)



รูปที่ 2.18 ขาสัญญาณและการเชื่อมต่อบน Keyes Relay 2R1B
ที่มา : http://www.thaieasyelec.net/archives/Manual/EFDV237/
EFDV237_2_Channels_Relay_Quick_Start_Guide.pdf

การทำงาน

ขาที่ 1 : +VCC ขาไฟ 5VDC

ขาที่ 2 : GND (ของรีเลย์)

ขาที่ 3 : ขาสัญญาณอินพุต Relay 1 (IN1)

ขาที่ 4 : ขาสัญญาณอินพุต Relay 2 (IN2)

ขาที่ 5 : COM (คอมมอนของ OPTO)

ขาที่ 6 : GND (กราวด์ของบอร์ดเป็นกราวด์เดียวกันกับขาที่ 2)

ขาที่ 7 : NC (Normal Close) ซึ่งหมายถึงหน้าสัมผัสแบบปกติปิด

ขาที่ 8 : COM (Common) ที่จะตัดหรือต่อวงจรจากขา NC, NO

ขาที่ 9 : NO (Normal Open) ซึ่งหมายถึงหน้าสัมผัสแบบปกติเปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 Power supply

2.6.1 Power supply C-360-12

Switching Power Supply 12 VDC 30A MODEL: C-360-12

Input Voltage: 100-120VAC, 8.0A /200-240VAC, 3.0A 50-60Hz

Output Voltage: 12V (DC), 30A

รูปที่ 2.19 Power supply C-360-12

ที่มา : http://www.dcelettrik.it/wp-content/uploads/2015/03/alim-24v-400w.jpg

2.7 Magnetic Lock

2.7.1 SP-KMG0600P Single Door Magnetic Lock

Lock Size: 250Lx47Wx25H (mm)

Armature plate: 180Lx38Wx11H (mm)

Holding Force: 280kg (600lbs)

Dual Voltage: DC12V/DC24V

Current Draw: DC12V/500mA DC24V/250mA

รูปที่ 2.20 Single Door Magnetic Lock

ที่มา : http://www.siamfinger.com/wp-content/uploads/2016/08-Magnetic-Lock.png

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

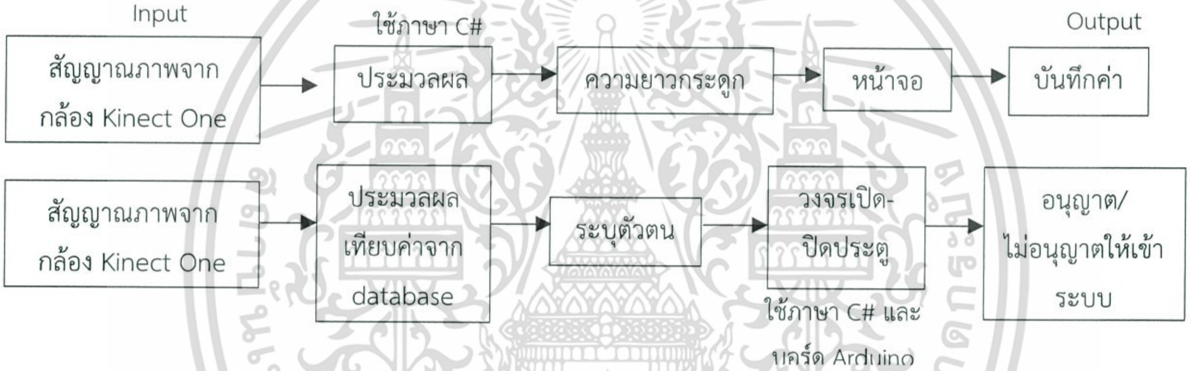
บทที่ 3

การออกแบบระบบ

3.1 หลักการออกแบบระบบ

3.1.1 บล็อกไดอะแกรมการทำงาน

การทำงานของระบบเก็บค่านี้ ใช้กล้อง Kinect ในการตรวจจับข้อต่อต่างๆ ในร่างกายของมนุษย์ จากนั้นจะทำการประมวลผลด้วย Microsoft Visual Studio 2013 เพื่อคำนวณค่าความยาวของส่วนต่างๆ บนร่างกาย และแสดงผลออกทางหน้าจอ จากนั้นทำการบันทึกค่าไว้ เมื่อผู้ใช้กลับมาอีกครั้ง ระบบระบุตัวตนจะทำการเปรียบเทียบกับค่าที่มีในฐานข้อมูล หากพบว่าค่าตรงกัน โปรแกรมจะสั่งการไปยังบอร์ด Arduino ซึ่งเชื่อมต่อกับระบบเปิด - ปิดประตู เพื่ออนุญาตให้ผู้ใช้งานระบบเข้าสู่ระบบได้ ดังแสดงในรูปที่ 3.1

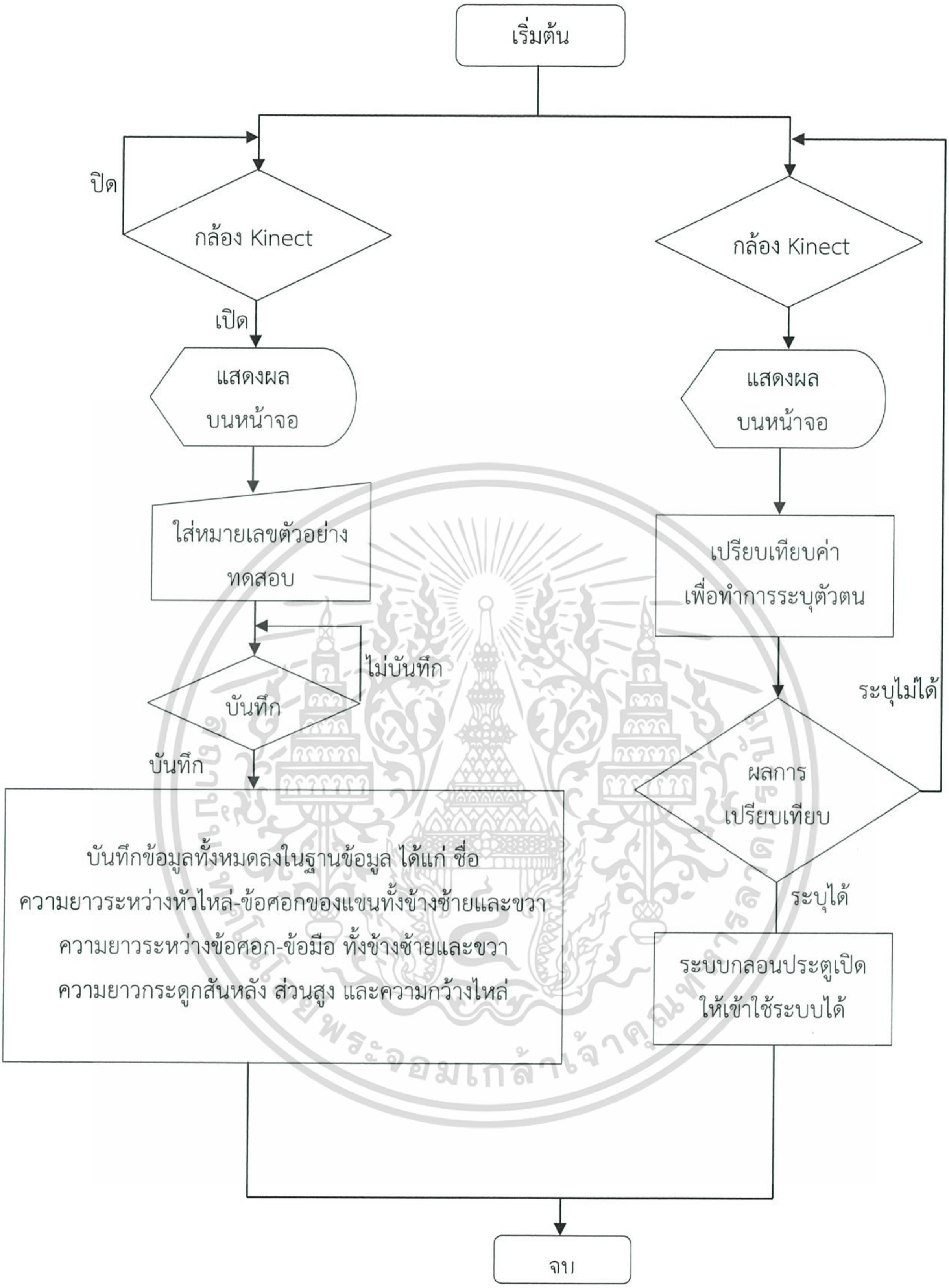


รูปที่ 3.1 บล็อกไดอะแกรมการทำงานของระบบ

3.1.2 แผนผังการทำงานของโปรแกรมหลัก

การทำงานของระบบ มีขั้นตอนการทำงานดังนี้ เริ่มต้นการทำงานของโปรแกรมโดยทำการตรวจสอบการเชื่อมต่อระหว่าง Kinect กับระบบ หากยังไม่ทำการเชื่อมต่อ ระบบจะไม่สามารถแสดงค่าใดๆ ได้ แต่หากทำการเชื่อมต่อสำเร็จระบบจะแสดงผลค่าสัดส่วนต่างๆ ออกทางหน้าจอ จากนั้นทำการใส่ชื่อและกดปุ่มเพื่อบันทึกค่า รวมทั้งสามารถประมวลผลเพื่อระบุตัวตนได้ โดยจะทำการเปรียบเทียบกับค่าที่บันทึกไว้ในฐานข้อมูล หากพบว่าค่าตรงกัน โปรแกรมจะสั่งการไปยังบอร์ด Arduino ซึ่งเชื่อมต่อกับระบบเปิด - ปิดประตู เพื่ออนุญาตให้ผู้ใช้งานระบบเข้าสู่ระบบได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แผนผังการทำงานของโปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 หลักการออกแบบระบบ

3.2.1 การทดลองที่ 1 การวัดความยาวร่างกายด้วย Kinect

1) กำหนดตำแหน่งที่จะติดตั้งกล้อง โดยในการทดสอบนี้ ผู้จัดทำได้กำหนดให้กลุ่มตัวอย่างยืนห่างจากกล้อง 2.15 เมตร และติดตั้งกล้อง โดยมีความสูงจากระนาบปกติ 1.25 เมตร

2) ให้กลุ่มตัวอย่างยืนในตำแหน่งที่กำหนด และแสดงท่าทางดังรูปที่ 3.3 จากนั้นทำการวัดค่าความยาวของร่างกายในส่วนที่ต้องการบันทึกค่า ได้แก่ ความยาวระหว่างหัวไหล่-ข้อศอก ความยาวระหว่างข้อศอก-ข้อมือ ความยาวกระดูกสันหลัง ส่วนสูง และความกว้างไหล่ ด้วยกล้อง Kinect

3) ทำการทดลองในข้อที่ 2 ทั้งหมด 10 ครั้ง และทำการบันทึกข้อมูลที่ได้ลงในฐานข้อมูล ในที่นี้คือโปรแกรมเสริม MySQL ซึ่งรวมอยู่ใน XAMPP

4) นำข้อมูลทั้งหมดจากข้อที่ 3 มาทำการคำนวณหาค่าเฉลี่ย และค่าส่วนเบี่ยงเบนมาตรฐาน บันทึกค่าที่คำนวณได้ลงในตารางที่ 4.1

รูปที่ 3.3 แสดงท่าทางที่ใช้ในการทดสอบ
ที่มา : ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 การทดลองที่ 2 การใช้ Kinect ระบุตัวตน

1) ให้กลุ่มตัวอย่างยืนในตำแหน่งที่กำหนด และแสดงท่าทางดังรูปที่ 3.3 จากนั้นทำการกดปุ่มเพื่อให้โปรแกรมทำการเปรียบเทียบค่าปัจจุบันกับค่าเดิมที่ได้บันทึกไว้

2) จากข้อที่ 1 เมื่อโปรแกรมตรวจเจอว่าค่าปัจจุบันมีค่าตรงกับค่าที่ได้บันทึกไว้ก่อนหน้านี้ จะทำการสั่งการไปยังบอร์ด Arduino เพื่อเปิดกลอนประตู

3) ทำการทดลองในข้อที่ 2 ทั้งหมด 10 ครั้ง และทำการบันทึกผลลัพธ์ที่ได้ลงในตารางที่ 4.2

4) นำข้อมูลทั้งหมดจากข้อที่ 3 มาทำการวิเคราะห์และสรุปผลการทดลอง

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

4.1 การทดลองที่ 1 การวัดค่าสัดส่วนร่างกายโดยใช้ Kinect ระบุตัวตน

ตารางที่ 4.1 แสดงความยาวสัดส่วนที่วัดค่าได้จากกล้อง Kinect

ตัวอย่างที่	ความยาวสัดส่วนข้างซ้าย		ความยาวสัดส่วนข้างขวา				
	ไหล่-ศอก	ศอก-ข้อมือ	ไหล่-ศอก	ศอก-ข้อมือ	สันหลัง	ส่วนสูง	ไหล่
1	28±0	23.4±0.52	27.2±0.42	28±0	62.8±0.42	174±0	33±0
2	26±0	24±0	24.3±0.48	26±0	60.8±0.48	167.4±0.52	35±0
3	24±0	21±0	25.4±0.52	24±0	55±0	153.4±0.52	33±0
4	26±0	25±0	26±0	26±0	55±0	166.2±0.42	34.2±0.42
5	26.6±0.52	23.6±0.52	27.7±0.48	26.4±0.52	56±0	164±0	34±0
6	29±0	25±0	29±0	29±0	60.1±0.32	171±0	35±0
7	26±0	24.3±0.48	28±0	26±0	61±0	167.6±0.84	35±0
8	27.1±0.57	24±0	27.3±0.48	26±0	60.5±0.47	169.6±0.84	34.4±0.52
9	27.8±0.42	25.1±0.56	28.2±0.42	27.3±0.52	62.4±0.52	172±0.47	35±0
10	25±0.47	23±0.47	26.7±0.48	25±0	57±0	161.4±0.52	33±0.47

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดลองที่ 2 การใช้ Kinect ระบุตัวตน

ตารางที่ 4.2 แสดงผลของการระบุตัวตน

ตัวอย่างที่	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	ครั้งที่ 6	ครั้งที่ 7	ครั้งที่ 8	ครั้งที่ 9	ครั้งที่ 10
1	✓	✓	✓	✓	x	✓	✓	✓	✓	✓
2	✓	✓	-	✓	✓	✓	✓	x	✓	✓
3	✓	x	✓	✓	✓	✓	✓	✓	✓	✓
4	✓	✓	✓	✓	x	✓	✓	✓	✓	-
5	✓	✓	✓	x	✓	✓	✓	✓	✓	✓
6	✓	✓	✓	-	✓	✓	✓	✓	✓	✓
7	✓	✓	x	✓	✓	✓	-	✓	✓	✓
8	x	✓	✓	✓	✓	✓	✓	✓	x	✓
9	✓	✓	-	✓	✓	✓	✓	✓	✓	✓
10	✓	✓	✓	✓	x	✓	x	✓	✓	✓

หมายเหตุ : ✓ แสดงถึง ระบุตัวตนถูกต้อง

x แสดงถึง ระบุตัวตนไม่ถูกต้อง

\- แสดงถึง ไม่สามารถระบุตัวตนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการทดลองและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

โครงงานนี้เป็นการศึกษาเกี่ยวกับการระบุตัวตนบุคคล โดยประยุกต์ใช้ Kinect ในการตรวจจับข้อต่อต่างๆ ของร่างกายมนุษย์ โดยโครงงานนี้จะทำการตรวจจับ หัวไหล่ ข้อศอก ข้อมือ สะโพก ศีรษะและเท้า เพื่อนำไปประมวลผลเป็นค่าความยาวของส่วนต่างๆ ของร่างกายเพื่อบันทึกลงในฐานข้อมูล นอกจากนี้ยังทำการเชื่อมต่อโปรแกรมเข้ากับบอร์ด Arduino เพื่อใช้สั่งการไปยังวงจรกลอนประตูสำหรับขั้นตอนการระบุตัวตน

จากผลการทดลองที่ 1 เมื่อทำการวัดสัดส่วนต่างๆ ของร่างกายด้วยกล้อง Kinect พบว่า ค่าความกว้างไหล่เป็นค่าที่มีความคลาดเคลื่อนน้อยที่สุด ซึ่งความคลาดเคลื่อนที่เกิดขึ้นนั้น เกิดจากกลุ่มตัวอย่างบางคนสวมใส่เสื้อผ้าที่หลวมโคร่งและสภาพแวดล้อมที่ไม่เอื้ออำนวยในบางสถานการณ์ ทำให้เกิดความผิดพลาดในการวัดสัดส่วนด้วยระบบ ส่งผลให้ค่ามีความคลาดเคลื่อนได้

จากผลการทดลองที่ 2 เมื่อใช้ Kinect ระบุตัวตน พบว่า ระบบสามารถระบุตัวตนได้ถูกต้องตามที่ออกแบบไว้ แต่มีข้อผิดพลาดเกิดขึ้นอยู่บ้าง โดยจากผลการทดลองนี้ เมื่อทำการทดลองระบุตัวตนทั้งหมด 10 คน คนละ 10 ครั้ง พบว่า มีความผิดพลาดทั้งหมด 15 ครั้ง จากทั้งหมด 100 ครั้ง คิดเป็น 15 เปอร์เซ็นต์

งานวิจัยนี้สามารถระบุตัวตนบุคคลได้จากการตรวจจับการเคลื่อนไหวของมนุษย์ โดยสามารถตรวจจับได้หลายข้อต่อในเวลาเดียวกัน และสามารถประมวลผลออกมาเป็นบุคคล โดยทำการวิเคราะห์ผ่านพารามิเตอร์ที่ใช้ คือ ความยาวระหว่างส่วนต่างๆ ในร่างกาย นอกจากนี้ยังสามารถลดปัญหาที่อาจเกิดจากระบบการระบุตัวตนชนิดอื่นๆ เช่น ลดโอกาสการส่งต่อเชื้อโรคจากบุคคลหนึ่งสู่อีกคน ปัญหาการระบุตัวตนผิดพลาดเนื่องจากลายนิ้วมือผิดเพี้ยน รวมทั้งลดปัญหาการเลียนแบบลายนิ้วมือ เนื่องจากโครงกระดูกนั้นเป็นเอกลักษณ์เฉพาะบุคคล ซึ่งสอดคล้องกับความต้องการในระบบรักษาความปลอดภัยที่มีคุณภาพที่อาจสูงขึ้นในอนาคต

5.2 ปัญหาและอุปสรรค

1. กลุ่มตัวอย่างสวมใส่เสื้อที่มีขนาดใหญ่และหลวมเกินไป ทำให้กล้องตรวจจับผิดพลาด
2. ในบริเวณที่ทำการทดลองมีคนรบกวนและวัตถุอื่นๆ จำนวนมาก ทำให้เมื่อทำการตรวจจับค่าด้วยกล้อง Kinect แล้วจึงเกิดความผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ข้อเสนอแนะ

1. ผู้ใช้ไม่ควรสวมใส่เสื้อผ้าที่มีขนาดใหญ่หรือหลวมมากเกินไป ซึ่งอาจส่งผลให้ค่าที่วัดได้มีความผิดพลาดเกิดขึ้น
2. พื้นที่ที่เหมาะสมในการใช้งานกล้อง Kinect คือบริเวณที่มิดชิด ปราศจากคนรบกวนหรือสิ่งของ ควรใช้ฉากปิดด้านหลัง เนื่องจากกล้องอาจตรวจจับวัตถุอื่น ทำให้ค่าที่ได้เกิดความคลาดเคลื่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] Langisser's Blog. **"หลักการทำงานและการประยุกต์ใช้งานของไคเนค(Kinect)"** [Online]. Available: https://langisser.wordpress.com/2011/03/07/. 2554.

[2] HONG'S ON GOOGLE+. **"Kinect"** [Online]. Available: https://kinectasia.wordpress.com/2012/04/04/kinect/

[3] ภูวดล ศิริกองธรรม. **"เทคนิคสำหรับการควบคุมอินเทอร์เน็ตทีวีโดยคีเนค ด้วยการผสมผสานระหว่างการรู้จำเสียงและการตรวจสอบการเคลื่อนไหว"** [Online]. Available: http://libdoc.dpu.ac.th/thesis/148604.pdf

[4] สุภาวดี เนตรนิล. **"การพัฒนาซอฟต์แวร์ด้วยโปรแกรมสำเร็จรูป"** [Online]. Available: http://61.19.212.44/~s53121supavadee/htdoc/Title.html

[5] ThaiEasyElec. **"บทความ Arduino คืออะไร?"** [Online]. Available: http://www.thaieasyelec.com/article-wiki/basic-electronics/%E0%B8%9A%E0%B8%97%E0%B8%84%E0%B8%A7%E0%B8%B2%E0%B8%A1-arduino-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-%E0%B9%80%E0%B8%A3%E0%B8%B4%E0%B9%88%E0%B8%A1%E0%B8%95%E0%B9%89%E0%B8%99%E0%B9%83%E0%B8%8A%E0%B9%89%E0%B8%87%E0%B8%B2%E0%B8%99-arduino.html

[6] psptech. **"รีเลย์ (Relay) คืออะไร?"** [Online]. Available: http://www.psptech.co.th/%E0%B8%A3%E0%B8%B5%E0%B9%80%E0%B8%A5%E0%B8%A2%E0%B9%8Crelay%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-15696.page

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โค้ดสำหรับบันทึกค่าสัดส่วนลงฐานข้อมูล

```
namespace Microsoft.Samples.Kinect.BodyBasics
{
    using System;
    using System.Collections.Generic;
    using System.ComponentModel;
    using System.Diagnostics;
    using System.Globalization;
    using System.IO;
    using System.Windows;
    using System.Windows.Media;
    using System.Windows.Media.Imaging;
    using Microsoft.Kinect;
    using MySql.Data.MySqlClient;


    public partial class MainWindow : Window, INotifyPropertyChanged
    {
        private const double HandSize = 30;

        private const double JointThickness = 3;

        private const double ClipBoundsThickness = 10;

        private const float InferredZPositionClamp = 0.1f;

        private readonly Brush handClosedBrush = new
SolidColorBrush(Color.FromArgb(128, 255, 0, 0));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
private readonly Brush handOpenBrush = new
SolidColorBrush(Color.FromArgb(128, 0, 255, 0));

private readonly Brush handLassoBrush = new
SolidColorBrush(Color.FromArgb(128, 0, 0, 255));

private readonly Brush trackedJointBrush = new
SolidColorBrush(Color.FromArgb(255, 68, 192, 68));

private readonly Brush inferredJointBrush = Brushes.Yellow;

private readonly Pen inferredBonePen = new Pen(Brushes.Gray, 1);

private DrawingGroup drawingGroup;

private DrawingImage imageSource;

private KinectSensor kinectSensor = null;

private CoordinateMapper coordinateMapper = null;

private BodyFrameReader bodyFrameReader = null;

private Body[] bodies = null;

private List<Tuple<JointType, JointType>> bones;

private int displayWidth;

private int displayHeight;

private List<Pen> bodyColors;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
private string statusText = null;

public MainWindow()
{

    this.kinectSensor = KinectSensor.GetDefault();
    this.coordinateMapper = this.kinectSensor.CoordinateMapper;

    FrameDescription frameDescription =
this.kinectSensor.DepthFrameSource.FrameDescription;

    this.displayWidth = frameDescription.Width;
    this.displayHeight = frameDescription.Height;

    this.bodyFrameReader = this.kinectSensor.BodyFrameSource.OpenReader();

    this.bones = new List<Tuple<JointType, JointType>>();
    this.bones.Add(new Tuple<JointType, JointType>(JointType.Head,
JointType.Neck));
    this.bones.Add(new Tuple<JointType, JointType>(JointType.Neck,
JointType.SpineShoulder));
    this.bones.Add(new Tuple<JointType, JointType>(JointType.SpineShoulder,
JointType.SpineMid));
    this.bones.Add(new Tuple<JointType, JointType>(JointType.SpineMid,
JointType.SpineBase));
    this.bones.Add(new Tuple<JointType, JointType>(JointType.SpineShoulder,
JointType.ShoulderRight));
    this.bones.Add(new Tuple<JointType, JointType>(JointType.SpineShoulder,
JointType.ShoulderLeft));
    this.bones.Add(new Tuple<JointType, JointType>(JointType.SpineBase,
JointType.HipRight));
    this.bones.Add(new Tuple<JointType, JointType>(JointType.SpineBase,
JointType.HipLeft));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
this.bones.Add(new Tuple<JointType, JointType>(JointType.ShoulderRight, JointType.ElbowRight));
this.bones.Add(new Tuple<JointType, JointType>(JointType.ElbowRight, JointType.WristRight));
this.bones.Add(new Tuple<JointType, JointType>(JointType.WristRight, JointType.HandRight));
this.bones.Add(new Tuple<JointType, JointType>(JointType.HandRight, JointType.HandTipRight));
this.bones.Add(new Tuple<JointType, JointType>(JointType.WristRight, JointType.ThumbRight));
this.bones.Add(new Tuple<JointType, JointType>(JointType.ShoulderLeft, JointType.ElbowLeft));
this.bones.Add(new Tuple<JointType, JointType>(JointType.ElbowLeft, JointType.WristLeft));
this.bones.Add(new Tuple<JointType, JointType>(JointType.WristLeft, JointType.HandLeft));
this.bones.Add(new Tuple<JointType, JointType>(JointType.HandLeft, JointType.HandTipLeft));
this.bones.Add(new Tuple<JointType, JointType>(JointType.WristLeft, JointType.ThumbLeft));
this.bones.Add(new Tuple<JointType, JointType>(JointType.HipRight, JointType.KneeRight));
this.bones.Add(new Tuple<JointType, JointType>(JointType.KneeRight, JointType.AnkleRight));
this.bones.Add(new Tuple<JointType, JointType>(JointType.AnkleRight, JointType.FootRight));
this.bones.Add(new Tuple<JointType, JointType>(JointType.HipLeft, JointType.KneeLeft));
this.bones.Add(new Tuple<JointType, JointType>(JointType.KneeLeft, JointType.AnkleLeft));
this.bones.Add(new Tuple<JointType, JointType>(JointType.AnkleLeft, JointType.FootLeft));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            this.bodyColors = new List<Pen>();

        this.bodyColors.Add(new Pen(Brushes.Red, 6));
        this.bodyColors.Add(new Pen(Brushes.Orange, 6));
        this.bodyColors.Add(new Pen(Brushes.Green, 6));
        this.bodyColors.Add(new Pen(Brushes.Blue, 6));
        this.bodyColors.Add(new Pen(Brushes.Indigo, 6));
        this.bodyColors.Add(new Pen(Brushes.Violet, 6));

        this.kinectSensor.IsAvailableChanged += this.Sensor_IsAvailableChanged;

        this.kinectSensor.Open();

        this.StatusText = this.kinectSensor.IsAvailable ?
Properties.Resources.RunningStatusText
                                                  : Properties.Resources.NoSensorStatusText;

        this.drawingGroup = new DrawingGroup();
        this.imageSource = new DrawingImage(this.drawingGroup);
        this.DataContext = this;

        this.InitializeComponent();
    }

    public event PropertyChangedEventHandler PropertyChanged;
    public ImageSource ImageSource
    {
        get
        {
            return this.imageSource;
        }
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
public string StatusText
{
    get
    {
        return this.statusText;
    }

    set
    {
        if (this.statusText != value)
        {
            this.statusText = value;

            if (this.PropertyChanged != null)
            {
                this.PropertyChanged(this, new
PropertyChangedEventArgs("StatusText"));
            }
        }
    }
}
private void MainWindow_Loaded(object sender, RoutedEventArgs e)
{
    if (this.bodyFrameReader != null)
    {
        this.bodyFrameReader.FrameArrived += this.Reader_FrameArrived;
    }
}

private void MainWindow_Closing(object sender, CancelEventArgs e)
{
    if (this.bodyFrameReader != null)
    {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        this.bodyFrameReader.Dispose();
        this.bodyFrameReader = null;
    }

    if (this.kinectSensor != null)
    {
        this.kinectSensor.Close();
        this.kinectSensor = null;
    }
}

private void Reader_FrameArrived(object sender, BodyFrameArrivedEventArgs e)
{
    bool dataReceived = false;

    using (BodyFrame bodyFrame = e.FrameReference.AcquireFrame())
    {
        if (bodyFrame != null)
        {
            if (this.bodies == null)
            {
                this.bodies = new Body[bodyFrame.BodyCount];
            }

            bodyFrame.GetAndRefreshBodyData(this.bodies);
            dataReceived = true;
        }
    }

    if (dataReceived)
    {
        using (DrawingContext dc = this.drawingGroup.Open())
        {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    dc.DrawRectangle(Brushes.Black, null, new Rect(0.0, 0.0,
this.displayWidth, this.displayHeight));

                    int penIndex = 0;
                    foreach (Body body in this.bodies)
                    {
                        Pen drawPen = this.bodyColors[penIndex++];

                        if (body.IsTracked)
                        {
                            this.DrawClippedEdges(body, dc);

                            IReadOnlyDictionary<JointType, Joint> joints = body.Joints;

                            Dictionary<JointType, Point> jointPoints = new
Dictionary<JointType, Point>();

                            foreach (JointType jointType in joints.Keys)
                            {

                                CameraSpacePoint position = joints[jointType].Position;
                                if (position.Z < 0)
                                {
                                    position.Z = InferredZPositionClamp;
                                }
                                //...............................................
                                Joint head = body.Joints[JointType.Head];
                                float headx = head.Position.X;
                                float heady = head.Position.Y;
                                float headz = head.Position.Z;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Joint shoulderRight = body.Joints[JointType.ShoulderRight];
float shoulderRightx = shoulderRight.Position.X;
float shoulderRighty = shoulderRight.Position.Y;
float shoulderRightz = shoulderRight.Position.Z;

Joint shoulderLeft = body.Joints[JointType.ShoulderLeft];
float shoulderLeftx = shoulderLeft.Position.X;
float shoulderLefty = shoulderLeft.Position.Y;
float shoulderLeftz = shoulderLeft.Position.Z;

Joint wristRight = body.Joints[JointType.WristRight];
float wristRightx = wristRight.Position.X;
float wristRighty = wristRight.Position.Y;
float wristRightz = wristRight.Position.Z;

Joint wristLeft = body.Joints[JointType.WristLeft];
float wristLeftx = wristLeft.Position.X;
float wristLefty = wristLeft.Position.Y;
float wristLeftz = wristLeft.Position.Z;

Joint spineBase = body.Joints[JointType.SpineBase];
float spineBasex = spineBase.Position.X;
float spineBasey = spineBase.Position.Y;
float spineBasez = spineBase.Position.Z;

Joint hipRight = body.Joints[JointType.HipRight];
float hipRightx = hipRight.Position.X;
float hipRighty = hipRight.Position.Y;
float hipRightz = hipRight.Position.Z;

Joint hipLeft = body.Joints[JointType.HipLeft];
float hipLeftx = hipLeft.Position.X;
float hipLefty = hipLeft.Position.Y;
float hipLeftz = hipLeft.Position.Z;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Joint footRight = body.Joints[JointType.FootRight];
float footRightx = footRight.Position.X;
float footRighty = footRight.Position.Y;
float footRightz = footRight.Position.Z;

Joint footLeft = body.Joints[JointType.FootLeft];
float footLeftx = footLeft.Position.X;
float footLefty = footLeft.Position.Y;
float footLeftz = footLeft.Position.Z;

// ส่วนสูง
float d1 = (heady - footRighty) / (heady - footRighty) * 100;
txt1.Text = d1.ToString("0");
// ความยาวสันหลัง
float d2 = (heady - spineBasey) / (heady - footRighty) * 100;
txt2.Text = d2.ToString("0");
// ความกว้างไหล่
float d3 = (shoulderRightx - shoulderLeftx) / (heady - footRighty) * 100;
txt3.Text = d3.ToString("0");
//ความยาวแขนขาว
float d4 = (shoulderRighty - wristRighty) / (heady - footRighty) * 100;
txt4.Text = d4.ToString("0");
//ความยาวแขนซ้าย
float d5 = (shoulderLefty - wristLefty) / (heady - footRighty) * 100;
txt5.Text = d5.ToString("0");
//ความยาวขาขวา
float d6 = (hipRighty - footRighty) / (heady - footRighty) * 100;
txt6.Text = d6.ToString("0");
//ความยาวขาซ้าย
float d7 = (hipLefty - footLefty) / (heady - footRighty) * 100;
txt7.Text = d7.ToString("0");

//................................................
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        DepthSpacePoint depthSpacePoint =
this.coordinateMapper.MapCameraPointToDepthSpace(position);
                        jointPoints[jointType] = new Point(depthSpacePoint.X,
depthSpacePoint.Y);
                    }

                    this.DrawBody(joints, jointPoints, dc, drawPen);

                    this.DrawHand(body.HandLeftState,
jointPoints[JointType.HandLeft], dc);
                    this.DrawHand(body.HandRightState,
jointPoints[JointType.HandRight], dc);
                }
            }

            this.drawingGroup.ClipGeometry = new RectangleGeometry(new
Rect(0.0, 0.0, this.displayWidth, this.displayHeight));
        }
    }
}

private void DrawBody(IReadOnlyDictionary<JointType, Joint> joints,
IDictionary<JointType, Point> jointPoints, DrawingContext drawingContext, Pen
drawingPen)
{

    foreach (var bone in this.bones)
    {
        this.DrawBone(joints, jointPoints, bone.Item1, bone.Item2, drawingContext,
drawingPen);
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            foreach (JointType jointType in joints.Keys)
            {
                Brush drawBrush = null;

                TrackingState trackingState = joints[jointType].TrackingState;

                if (trackingState == TrackingState.Tracked)
                {
                    drawBrush = this.trackedJointBrush;
                }
                else if (trackingState == TrackingState.Inferred)
                {
                    drawBrush = this.inferredJointBrush;
                }
                if (drawBrush != null)
                {
                    drawingContext.DrawEllipse(drawBrush, null, jointPoints[jointType],
JointThickness, JointThickness);
                }
            }
        }
        private void DrawBone(IReadOnlyDictionary<JointType, Joint> joints,
IDictionary<JointType, Point> jointPoints, JointType jointType0, JointType jointType1,
DrawingContext drawingContext, Pen drawingPen)
        {
            Joint joint0 = joints[jointType0];
            Joint joint1 = joints[jointType1];

            if (joint0.TrackingState == TrackingState.NotTracked ||
                joint1.TrackingState == TrackingState.NotTracked)
            {
                return;
            }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Pen drawPen = this.inferredBonePen;
            if ((joint0.TrackingState == TrackingState.Tracked) && (joint1.TrackingState ==
TrackingState.Tracked))
            {
                drawPen = drawingPen;
            }

            drawingContext.DrawLine(drawPen, jointPoints[jointType0],
jointPoints[jointType1]);
        }

        private void DrawHand(HandState handState, Point handPosition,
DrawingContext drawingContext)
        {
            switch (handState)
            {
                case HandState.Closed:
                    drawingContext.DrawEllipse(this.handClosedBrush, null, handPosition,
HandSize, HandSize);
                    break;

                case HandState.Open:
                    drawingContext.DrawEllipse(this.handOpenBrush, null, handPosition,
HandSize, HandSize);
                    break;

                case HandState.Lasso:
                    drawingContext.DrawEllipse(this.handLassoBrush, null, handPosition,
HandSize, HandSize);
                    break;
            }
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
private void DrawClippedEdges(Body body, DrawingContext drawingContext)
{
    FrameEdges clippedEdges = body.ClippedEdges;

    if (clippedEdges.HasFlag(FrameEdges.Bottom))
    {
        drawingContext.DrawRectangle(
            Brushes.Red,
            null,
            new Rect(0, this.displayHeight - ClipBoundsThickness, this.displayWidth,
ClipBoundsThickness));
    }
    if (clippedEdges.HasFlag(FrameEdges.Top))
    {
        drawingContext.DrawRectangle(
            Brushes.Red,
            null,
            new Rect(0, 0, this.displayWidth, ClipBoundsThickness));
    }
    if (clippedEdges.HasFlag(FrameEdges.Left))
    {
        drawingContext.DrawRectangle(
            Brushes.Red,
            null,
            new Rect(0, 0, ClipBoundsThickness, this.displayHeight));
    }
    if (clippedEdges.HasFlag(FrameEdges.Right))
    {
        drawingContext.DrawRectangle(
            Brushes.Red,
            null,
            new Rect(this.displayWidth - ClipBoundsThickness, 0,
ClipBoundsThickness, this.displayHeight));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }
        }
        private void Sensor_IsAvailableChanged(object sender,
IsAvailableChangedEventArgs e)
        {
            // on failure, set the status text
            this.StatusText = this.kinectSensor.IsAvailable ?
Properties.Resources.RunningStatusText
                                                            :
Properties.Resources.SensorNotAvailableStatusText;
        }

        private void Button_Click(object sender, RoutedEventArgs e)
        {
            string sql = "insert into B  (name,d1,d2,d3,d4,d5,d6,d7) values('" +
this.name.Text + "' , '" + this.txt1.Text + "' , '" + this.txt2.Text + "', '" + this.txt3.Text +
"' , '" + this.txt4.Text + "' , '" + this.txt5.Text + "' , '" + this.txt6.Text + "' , '" +
this.txt7.Text + "' )  ";



            MySqlConnection con = new
MySqlConnection("SERVER=127.0.0.1;DATABASE=test");
            MySqlCommand cmd = new MySqlCommand(sql, con);
            con.Open();

            cmd.ExecuteNonQuery();

        }


    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข
โค้ดสำหรับระบุตัวตน

```
namespace Microsoft.Samples.Kinect.BodyBasics
{
    using System;
    using System.Collections.Generic;
    using System.ComponentModel;
    using System.Diagnostics;
    using System.Globalization;
    using System.IO;
    using System.Windows;
    using System.Windows.Media;
    using System.Windows.Media.Imaging;
    using Microsoft.Kinect;
    using MySql.Data.MySqlClient;
    using System.IO.Ports;
    using System.IO;

    public partial class MainWindow : Window, INotifyPropertyChanged
    {

        SerialPort mySerialPort = new SerialPort("COM6", 115200, Parity.None, 8,
StopBits.One);

        private const double HandSize = 30;

        private const double JointThickness = 3;

        private const double ClipBoundsThickness = 10;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
private const float InferredZPositionClamp = 0.1f;

private readonly Brush handClosedBrush = new
SolidColorBrush(Color.FromArgb(128, 255, 0, 0));

private readonly Brush handOpenBrush = new
SolidColorBrush(Color.FromArgb(128, 0, 255, 0));

private readonly Brush handLassoBrush = new
SolidColorBrush(Color.FromArgb(128, 0, 0, 255));

private readonly Brush trackedJointBrush = new
SolidColorBrush(Color.FromArgb(255, 68, 192, 68));

private readonly Brush inferredJointBrush = Brushes.Yellow;

private readonly Pen inferredBonePen = new Pen(Brushes.Gray, 1);

private DrawingGroup drawingGroup;

private DrawingImage imageSource;

private KinectSensor kinectSensor = null;

private CoordinateMapper coordinateMapper = null;

private BodyFrameReader bodyFrameReader = null;

private Body[] bodies = null;

private List<Tuple<JointType, JointType>> bones;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
private int displayWidth;

private int displayHeight;

private List<Pen> bodyColors;

private string statusText = null;

public MainWindow()
{

    this.InitializeComponent();

    this.mySerialPort.Open();

    this.kinectSensor = KinectSensor.GetDefault();

    this.coordinateMapper = this.kinectSensor.CoordinateMapper;

    FrameDescription frameDescription =
this.kinectSensor.DepthFrameSource.FrameDescription;

    this.displayWidth = frameDescription.Width;
    this.displayHeight = frameDescription.Height;
    this.bodyFrameReader = this.kinectSensor.BodyFrameSource.OpenReader();

    this.bones = new List<Tuple<JointType, JointType>>();

    this.bones.Add(new Tuple<JointType, JointType>(JointType.Head,
JointType.Neck));
    this.bones.Add(new Tuple<JointType, JointType>(JointType.Neck,
JointType.SpineShoulder));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        this.bones.Add(new Tuple<JointType, JointType>(JointType.SpineShoulder,
JointType.SpineMid));
        this.bones.Add(new Tuple<JointType, JointType>(JointType.SpineMid,
JointType.SpineBase));
        this.bones.Add(new Tuple<JointType, JointType>(JointType.SpineShoulder,
JointType.ShoulderRight));
        this.bones.Add(new Tuple<JointType, JointType>(JointType.SpineShoulder,
JointType.ShoulderLeft));
        this.bones.Add(new Tuple<JointType, JointType>(JointType.SpineBase,
JointType.HipRight));
        this.bones.Add(new Tuple<JointType, JointType>(JointType.SpineBase,
JointType.HipLeft));
        this.bones.Add(new Tuple<JointType, JointType>(JointType.ShoulderRight,
JointType.ElbowRight));
        this.bones.Add(new Tuple<JointType, JointType>(JointType.ElbowRight,
JointType.WristRight));
        this.bones.Add(new Tuple<JointType, JointType>(JointType.WristRight,
JointType.HandRight));
        this.bones.Add(new Tuple<JointType, JointType>(JointType.HandRight,
JointType.HandTipRight));
        this.bones.Add(new Tuple<JointType, JointType>(JointType.WristRight,
JointType.ThumbRight));
        this.bones.Add(new Tuple<JointType, JointType>(JointType.ShoulderLeft,
JointType.ElbowLeft));
        this.bones.Add(new Tuple<JointType, JointType>(JointType.ElbowLeft,
JointType.WristLeft));
        this.bones.Add(new Tuple<JointType, JointType>(JointType.WristLeft,
JointType.HandLeft));
        this.bones.Add(new Tuple<JointType, JointType>(JointType.HandLeft,
JointType.HandTipLeft));
        this.bones.Add(new Tuple<JointType, JointType>(JointType.WristLeft,
JointType.ThumbLeft));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
this.bones.Add(new Tuple<JointType, JointType>(JointType.HipRight,
JointType.KneeRight));
this.bones.Add(new Tuple<JointType, JointType>(JointType.KneeRight,
JointType.AnkleRight));
this.bones.Add(new Tuple<JointType, JointType>(JointType.AnkleRight,
JointType.FootRight));
this.bones.Add(new Tuple<JointType, JointType>(JointType.HipLeft,
JointType.KneeLeft));
this.bones.Add(new Tuple<JointType, JointType>(JointType.KneeLeft,
JointType.AnkleLeft));
this.bones.Add(new Tuple<JointType, JointType>(JointType.AnkleLeft,
JointType.FootLeft));

this.bodyColors = new List<Pen>();

this.bodyColors.Add(new Pen(Brushes.Red, 6));
this.bodyColors.Add(new Pen(Brushes.Orange, 6));
this.bodyColors.Add(new Pen(Brushes.Green, 6));
this.bodyColors.Add(new Pen(Brushes.Blue, 6));
this.bodyColors.Add(new Pen(Brushes.Indigo, 6));
this.bodyColors.Add(new Pen(Brushes.Violet, 6));

this.kinectSensor.IsAvailableChanged += this.Sensor_IsAvailableChanged;

this.kinectSensor.Open();

this.StatusText = this.kinectSensor.IsAvailable ?
Properties.Resources.RunningStatusText
                                : Properties.Resources.NoSensorStatusText;

this.drawingGroup = new DrawingGroup();

this.imageSource = new DrawingImage(this.drawingGroup);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        this.DataContext = this;

        this.InitializeComponent();
    }

    public event PropertyChangedEventHandler PropertyChanged;

    public ImageSource ImageSource
    {
        get
        {
            return this.imageSource;
        }
    }

    public string StatusText
    {
        get
        {
            return this.statusText;
        }
        set
        {
            if (this.statusText != value)
            {
                this.statusText = value;

                if (this.PropertyChanged != null)
                {
                    this.PropertyChanged(this, new
PropertyChangedEventArgs("StatusText"));
                }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
    }
}

private void MainWindow_Loaded(object sender, RoutedEventArgs e)
{
    if (this.bodyFrameReader != null)
    {
        this.bodyFrameReader.FrameArrived += this.Reader_FrameArrived;
    }
}
private void MainWindow_Closing(object sender, CancelEventArgs e)
{
    if (this.bodyFrameReader != null)
    {
        this.bodyFrameReader.Dispose();
        this.bodyFrameReader = null;
    }
    if (this.kinectSensor != null)
    {
        this.kinectSensor.Close();
        this.kinectSensor = null;
    }
}

private void Reader_FrameArrived(object sender, BodyFrameArrivedEventArgs e)
{
    bool dataReceived = false;

    using (BodyFrame bodyFrame = e.FrameReference.AcquireFrame())
    {
        if (bodyFrame != null)
        {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if (this.bodies == null)
                {
                    this.bodies = new Body[bodyFrame.BodyCount];
                }

                bodyFrame.GetAndRefreshBodyData(this.bodies);
                dataReceived = true;
            }
        }

        if (dataReceived)
        {
            using (DrawingContext dc = this.drawingGroup.Open())
            {

                dc.DrawRectangle(Brushes.Black, null, new Rect(0.0, 0.0,
this.displayWidth, this.displayHeight));

                int penIndex = 0;
                foreach (Body body in this.bodies)
                {
                    Pen drawPen = this.bodyColors[penIndex++];

                    if (body.IsTracked)
                    {
                        this.DrawClippedEdges(body, dc);
                        IReadOnlyDictionary<JointType, Joint> joints = body.Joints;

                        Dictionary<JointType, Point> jointPoints = new
Dictionary<JointType, Point>();

                        foreach (JointType jointType in joints.Keys)
                        {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CameraSpacePoint position = joints[jointType].Position;
if (position.Z < 0)
{
    position.Z = InferredZPositionClamp;
}

Joint head = body.Joints[JointType.Head];
float headx = head.Position.X;
float heady = head.Position.Y;
float headz = head.Position.Z;

Joint shoulderRight = body.Joints[JointType.ShoulderRight];
float shoulderRightx = shoulderRight.Position.X;
float shoulderRighty = shoulderRight.Position.Y;
float shoulderRightz = shoulderRight.Position.Z;

Joint shoulderLeft = body.Joints[JointType.ShoulderLeft];
float shoulderLeftx = shoulderLeft.Position.X;
float shoulderLefty = shoulderLeft.Position.Y;
float shoulderLeftz = shoulderLeft.Position.Z;

Joint wristRight = body.Joints[JointType.WristRight];
float wristRightx = wristRight.Position.X;
float wristRighty = wristRight.Position.Y;
float wristRightz = wristRight.Position.Z;

Joint wristLeft = body.Joints[JointType.WristLeft];
float wristLeftx = wristLeft.Position.X;
float wristLefty = wristLeft.Position.Y;
float wristLeftz = wristLeft.Position.Z;

Joint spineBase = body.Joints[JointType.SpineBase];
float spineBasex = spineBase.Position.X;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
float spineBasey = spineBase.Position.Y;
float spineBasez = spineBase.Position.Z;

Joint hipRight = body.Joints[JointType.HipRight];
float hipRightx = hipRight.Position.X;
float hipRighty = hipRight.Position.Y;
float hipRightz = hipRight.Position.Z;

Joint hipLeft = body.Joints[JointType.HipLeft];
float hipLeftx = hipLeft.Position.X;
float hipLefty = hipLeft.Position.Y;
float hipLeftz = hipLeft.Position.Z;

Joint footRight = body.Joints[JointType.FootRight];
float footRightx = footRight.Position.X;
float footRighty = footRight.Position.Y;
float footRightz = footRight.Position.Z;

Joint footLeft = body.Joints[JointType.FootLeft];
float footLeftx = footLeft.Position.X;
float footLefty = footLeft.Position.Y;
float footLeftz = footLeft.Position.Z;

// ส่วนสูง
float d1 = (heady - footRighty) / (heady - footRighty) * 100;
text1.Text = d1.ToString("0");
// ความยาวสันหลัง
float d2 = (heady - spineBasey) / (heady - footRighty) * 100;
text2.Text = d2.ToString("0");
// ความกว้างไหล่
float d3 = (shoulderRightx - shoulderLeftx) / (heady - footRighty) * 100;
text3.Text = d3.ToString("0");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//ความยาวแขนขาว
float d4 = (shoulderRighty - wristRighty) / (heady - footRighty) * 100;
text4.Text = d4.ToString("0");
//ความยาวแขนซ้าย
float d5 = (shoulderLefty - wristLefty) / (heady - footRighty) * 100;
text5.Text = d5.ToString("0");
//ความยาวขาขวา
float d6 = (hipRighty - footRighty) / (heady - footRighty) * 100;
text6.Text = d6.ToString("0");
//ความยาวขาซ้าย
float d7 = (hipLefty - footLefty) / (heady - footRighty) * 100;
text7.Text = d7.ToString("0");

        //..........................................................................................................
        DepthSpacePoint depthSpacePoint =
this.coordinateMapper.MapCameraPointToDepthSpace(position);
        jointPoints[jointType] = new Point(depthSpacePoint.X,
depthSpacePoint.Y);
      }

      this.DrawBody(joints, jointPoints, dc, drawPen);

      this.DrawHand(body.HandLeftState,
jointPoints[JointType.HandLeft], dc);
      this.DrawHand(body.HandRightState,
jointPoints[JointType.HandRight], dc);
    }
  }
  this.drawingGroup.ClipGeometry = new RectangleGeometry(new
Rect(0.0, 0.0, this.displayWidth, this.displayHeight));
 }
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        private void DrawBody(IReadOnlyDictionary<JointType, Joint> joints,
IDictionary<JointType, Point> jointPoints, DrawingContext drawingContext, Pen
drawingPen)
        {

            foreach (var bone in this.bones)
            {
                this.DrawBone(joints, jointPoints, bone.Item1, bone.Item2, drawingContext,
drawingPen);
            }

            foreach (JointType jointType in joints.Keys)
            {
                Brush drawBrush = null;

                TrackingState trackingState = joints[jointType].TrackingState;

                if (trackingState == TrackingState.Tracked)
                {
                    drawBrush = this.trackedJointBrush;
                }
                else if (trackingState == TrackingState.Inferred)
                {
                    drawBrush = this.inferredJointBrush;
                }

                if (drawBrush != null)
                {
                    drawingContext.DrawEllipse(drawBrush, null, jointPoints[jointType],
JointThickness, JointThickness);
                }
            }
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        private void DrawBone(IReadOnlyDictionary<JointType, Joint> joints,
    IDictionary<JointType, Point> jointPoints, JointType jointType0, JointType jointType1,
    DrawingContext drawingContext, Pen drawingPen)
        {
            Joint joint0 = joints[jointType0];
            Joint joint1 = joints[jointType1];

            if (joint0.TrackingState == TrackingState.NotTracked ||
                joint1.TrackingState == TrackingState.NotTracked)
            {
                return;
            }

            Pen drawPen = this.inferredBonePen;
            if ((joint0.TrackingState == TrackingState.Tracked) && (joint1.TrackingState ==
    TrackingState.Tracked))
            {
                drawPen = drawingPen;
            }

            drawingContext.DrawLine(drawPen, jointPoints[jointType0],
    jointPoints[jointType1]);
        }

        private void DrawHand(HandState handState, Point handPosition,
    DrawingContext drawingContext)
        {
            switch (handState)
            {
                case HandState.Closed:
                    drawingContext.DrawEllipse(this.handClosedBrush, null, handPosition,
    HandSize, HandSize);
                    break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                case HandState.Open:
                    drawingContext.DrawEllipse(this.handOpenBrush, null, handPosition,
HandSize, HandSize);
                    break;

                case HandState.Lasso:
                    drawingContext.DrawEllipse(this.handLassoBrush, null, handPosition,
HandSize, HandSize);
                    break;
            }
        }

        private void DrawClippedEdges(Body body, DrawingContext drawingContext)
        {
            FrameEdges clippedEdges = body.ClippedEdges;

            if (clippedEdges.HasFlag(FrameEdges.Bottom))
            {
                drawingContext.DrawRectangle(
                    Brushes.Red,
                    null,
                    new Rect(0, this.displayHeight - ClipBoundsThickness, this.displayWidth,
ClipBoundsThickness));
            }
            if (clippedEdges.HasFlag(FrameEdges.Top))
            {
                drawingContext.DrawRectangle(
                    Brushes.Red,
                    null,
                    new Rect(0, 0, this.displayWidth, ClipBoundsThickness));
            }
            if (clippedEdges.HasFlag(FrameEdges.Left))
            {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            drawingContext.DrawRectangle(
                Brushes.Red,
                null,
                new Rect(0, 0, ClipBoundsThickness, this.displayHeight));
        }

        if (clippedEdges.HasFlag(FrameEdges.Right))
        {
            drawingContext.DrawRectangle(
                Brushes.Red,
                null,
                new Rect(this.displayWidth - ClipBoundsThickness, 0,
ClipBoundsThickness, this.displayHeight));
    }

    private void Sensor_IsAvailableChanged(object sender,
IsAvailableChangedEventArgs e)
    {
        this.StatusText = this.kinectSensor.IsAvailable ?
Properties.Resources.RunningStatusText
                                                    :
Properties.Resources.SensorNotAvailableStatusText;
    }
    private void Button_Click(object sender, RoutedEventArgs e)
    {
        {
            string sql = "select * from B where d1 ='" + this.text1.Text + "' and d2 = '"
+ this.text2.Text + "' and d3 = '" + this.text3.Text + "'and d4 = '" + this.text4.Text + "'
and d5 = '" + this.text5.Text + "'  and d6 = '" + this.text6.Text + "' and d7 = '" +
this.text7.Text + "'";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MySqlConnection con = new
MySqlConnection("SERVER=127.0.0.1;DATABASE=test");
                MySqlCommand cmd = new MySqlCommand(sql, con);

                con.Open();

                MySqlDataReader mid = cmd.ExecuteReader();

                if (mid.Read())
                {
                    mySerialPort.Write("1");
                    string sname = mid.GetString(0);
                    name_txt.Text = sname;
                }
                else

                  mySerialPort.Write("0");

                System.Threading.Thread.Sleep(500);



            }
        }
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค
โค้ดสำหรับควบคุมรีเลย์

```
int sw = 8;
int led= 9;
void setup()
 {
  Serial.begin(115200);
   pinMode(sw, OUTPUT);
 digitalWrite(sw, LOW);


 }

 void loop() {



 if ( Serial.available()) {
   char s = Serial.read();
    if (s=='1') {
    digitalWrite(sw, HIGH);
    delay(3000);
    pinMode(led, OUTPUT);



    }





  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้