



ระบบบันทึกเวลาด้วยนิ้วมือ

นายณัฏวัฒน์ ศรีสูงค์

รหัสนักศึกษา 613040320-5

นางสาวณัฐธิดา ม่วงทอง

รหัสนักศึกษา 613040470-6

รายงานนี้เป็นรายงานโครงการของนักศึกษาซึ่งเสนอเป็นส่วนหนึ่งในหลักสูตรวิศวกรรม
ศาสตรบัณฑิต (วิศวกรรมระบบอิเล็กทรอนิกส์)

สาขาวิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น
ปีการศึกษา 2564



Fingerprint Attendance System

Nantawat srisuwong

Student ID 613040320-5

Natthida Moungtong

Student ID 613040470-6

**This is the report of the students' project assignment submitted in partial fulfillment of
the requirement for the degree of Bachelor of Engineering
(Electronic Systems Engineering)**

**Department of Electrical Engineering
Faculty of Engineering, Khon Kaen University
Academic Year 2021**

ใบประเมินผลงาน

ชื่อเรื่องภาษาไทย ระบบบันทึกเวลาด้วยนิ้วมือ

ชื่อเรื่องภาษาอังกฤษ Fingerprint Attendance System

ผู้จัดทำ

นายนั้นทวัฒน์ ศรีสูงค์

รหัสนักศึกษา 613040320-5

นางสาวณัฐริดา ม่วงทอง

รหัสนักศึกษา 613040470-6

อาจารย์ที่ปรึกษา

(อาจารย์สฤติพร พรนิมิตร)

อาจารย์ผู้ร่วมประเมิน

(.....)

(.....)

ประเมินผล ณ วันที่ **XX**

กิตติกรรมประกาศ

โครงการฉบับนี้เสร็จสมบูรณ์เป็นอย่างดีได้ด้วยความช่วยเหลือ และการให้คำปรึกษาจาก อาจารย์สถิรพร พรนิมิตร อาจารย์ที่ปรึกษาโครงการ ที่กรุณาให้คำแนะนำตลอดจนปรับปรุงแก้ไข ข้อบกพร่องต่าง ๆ ทุกขั้นตอนที่ได้ทำการศึกษา ทั้งการวางแผนโครงการ กระบวนการออกแบบทาง วิศวกรรม การตรวจสอบและแก้ไขงาน รวมทั้งรายงานเล่มนี้

ขอขอบคุณคณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น อันเป็นสถานที่ศึกษาที่ประสิทธิ ประสิทธิภาพความรู้ ขอกราบขอบคุณคุณคณาจารย์ประจำคณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น ทุกท่าน ที่ได้ให้ความรู้สั่งสอนรายวิชาพื้นที่ในการศึกษา ให้คำแนะนำ และข้อคิดเห็นอันเป็นประโยชน์ แก่โครงการฉบับนี้ ขอขอบคุณเจ้าหน้าที่และบุคลากรคณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น ที่ได้ให้ความช่วยเหลือและอำนวยความสะดวกในด้านต่าง ๆ

กราบขอบพระคุณบิดามารดา ครอบครัว และเพื่อนทุกคน ที่สนับสนุนและให้กำลังใจจน โครงการฉบับนี้สำเร็จด้วยดี คุณค่าและประโยชน์อันพึงมีจากการศึกษาโครงการนี้ ขอโน้มบงกช พระคุณบิดามารดาและบูรพาจารย์ทุกท่านที่ได้อบรมสั่งสอน และให้ความรู้มาโดยตลอด ทำให้ การศึกษาโครงการฉบับนี้สำเร็จลงได้

นายณัฏฐวัฒน์ ศรีสูงศักดิ์
นางสาวณัฐริดา ม่วงทอง

บทคัดย่อ

ปัจจุบันมีการนำเซนเซอร์เข้ามามีบทบาทในชีวิตประจำวันมากขึ้น รวมไปถึงเซนเซอร์ในด้านการตรวจสอบความถูกต้อง เพื่อเพิ่มประสิทธิภาพและความรวดเร็วในการทำงาน เซนเซอร์ที่นิยมใช้กันแพร่หลายอีกชนิดคือ เซนเซอร์สแกนลายนิ้วมือ เนื่องจากลายนิ้วมือเป็นเอกลักษณ์เฉพาะตัวที่ไม่สามารถลอกเลียนแบบหรือปลอมแปลงได้ จึงมีการนำมาใช้ในการตรวจสอบอย่างแพร่หลายมากขึ้นในปัจจุบัน โครงการฉบับนี้ได้มีการนำการสแกนลายนิ้วมือมาใช้ในระบบบันทึกเวลาโดยใช้ Optical Sensor และภาษาโปรแกรมไพธอน ในการควบคุมการทำงานของเซนเซอร์สแกนลายนิ้วมือ โดยภายในโปรแกรมสามารถบันทึกลายนิ้วมือได้สูงสุด 127 ผู้ใช้ โดยข้อมูลจะถูกจัดเก็บไว้ในหน่วยความจำของเซนเซอร์ การทำงานของเซนเซอร์สามารถค้นหาลายนิ้วมือที่ถูกบันทึกไว้ก่อนหน้านี้ และสามารถลบลายนิ้วมือที่บันทึกไว้แล้ว มีการจัดการฐานข้อมูลโดยใช้ SQL สามารถรับข้อมูลเซนเซอร์และส่งข้อมูลผ่านระบบเครือข่ายไร้สายไปยังผู้ใช้งานได้ โดยที่เซนเซอร์แต่ละตัวจะเชื่อมต่อไปยังฐานข้อมูลเดียวกัน และมีการสร้างสัญลักษณ์แทนการพิมพ์คำสั่งในการสั่งการของผู้ใช้งานเพื่อความสะดวกและรวดเร็วในการใช้งาน

Abstract

Nowadays, sensors are increasingly playing a role in daily life. Including sensors for authentication to increase efficiency and speed of work. Another widely used type of sensor is the fingerprint sensor. Because fingerprints are unique and cannot be copied or falsified. Therefore, it has been used in the examination more widely today. In this project, fingerprint scanning is implemented in a time recording system using Optical sensors and Python programming language. To control the operation of the fingerprint sensor. Within the program, up to 127 users can register fingerprints, and the data is stored in the sensor's memory. The function of the sensor can search for fingerprints that have been recorded previously and can delete saved fingerprints. The database is managed by SQL. Sensor data can be obtained and transmitted over a wireless network to the user. where each sensor is connected to the same database and has created a symbol instead of typing commands in the user's command for convenience and speed for use.

สารบัญ

หน้า

บทคัดย่อ.....	ก
Abstract.....	ข
สารบัญ	ค
สารบัญรูป	ง
คำอธิบายสัญลักษณ์และคำย่อ	จ
บทที่ 1 บทนำ	6
1.1 ประวัติความเป็นมา	6
บทที่ 2 เอกสารที่เกี่ยวข้อง.....	8
2.1 Raspberry Pi.....	8
บทที่ 3 วิธีการดำเนินการ	14
3.1 Block Diagram ของระบบ	14
3.2 หลักการทำงานของโปรแกรมไพธอน.....	15
3.3.1. การเชื่อมต่อ USB to Serial port.....	16
3.3.2 การใช้โปรแกรมภาษาไพธอน.....	17
3.3.3 สร้างฐานข้อมูล	17
บทที่ 4 การประเมินผล	18
4.1 การลงทะเบียนลายนิ้วมือ	18
4.2 การค้นหาลายนิ้วที่ลงทะเบียนไว้	20
4.3 การลบลายนิ้วมือ	21
4.4 การใช้ฐานข้อมูล.....	22
บทที่ 5 สรุปและอภิปราย	24
เอกสารอ้างอิง	26
ภาคผนวก	i
ภาคผนวก ก โปรแกรมภาษาไพธอน	ii
ภาคผนวก ข	vii
ภาคผนวก ข ใช้งาน Fingerprint ร่วมกับโปรแกรม SFG Demo V.2.0.....	vii

สารบัญรูป

รูปที่ 1	แผนผังแสดงการทำงานของระบบ	6
รูปที่ 2	Raspberry Pi	8
รูปที่ 3	ส่วนประกอบของบอร์ด Raspberry Pi	9
รูปที่ 4	การทำงานของ Optical Sensor.....	10
รูปที่ 5	การทำงานของ Optical Sensor.....	11
รูปที่ 6	ตัวอย่างไอคอนที่ใช้ในการเรียกเปิดแอปพลิเคชัน.....	Error! Bookmark not defined.
รูปที่ 7	แผนผังแสดงการทำงานของโปรแกรมไพธอน	15
รูปที่ 8	USB to Serial port.....	16
รูปที่ 9	แสดงผล windows ไม่สามารถเชื่อมต่อกับ USB ได้	16
รูปที่ 10	การต่อสายจากเซนเซอร์เข้า USB	17
รูปที่ 11	แสดงผลหน้าโปรแกรม	18
รูปที่ 12	คำสั่งโปรแกรมไพธอนสำหรับเลือกโหมดการทำงานของโปรแกรม.....	18
รูปที่ 13	แสดงผลเมื่อเรียกใช้คำสั่ง e.....	19
รูปที่ 14	แสดงผลเมื่อบันทึกถายนิ้วมือสำเร็จ	19
รูปที่ 15	คำสั่งโปรแกรมไพธอนสำหรับเก็บถายนิ้วมือ	20
รูปที่ 16	แสดงผลเมื่อเรียกใช้คำสั่ง f ในการค้นหาถายนิ้วมือที่บันทึกไว้ก่อนหน้า	20
รูปที่ 17	คำสั่งโปรแกรมไพธอนสำหรับค้นหาถายนิ้วมือ	20
รูปที่ 18	แสดงผลเมื่อใช้คำสั่ง f ค้นหาถายนิ้วมือไม่สำเร็จ	21
รูปที่ 19	แสดงผลเมื่อใช้คำสั่ง d ในการลบถายนิ้วมือ	21
รูปที่ 20	ไฟล์ข้อมูล .csv ที่เก็บข้อมูลรหัสนักศึกษา.....	22
รูปที่ 21	เลือก Port สำหรับเชื่อมต่อกับ Fingerprint	vii
รูปที่ 22	เลือก Port สำหรับเชื่อมต่อกับ Fingerprint	viii
รูปที่ 23	กำหนด Address ของถายนิ้วมือ.....	viii
รูปที่ 24	แสดงผลลัพท์เมื่อพบถายนิ้วมือที่บันทึกไว้ก่อนหน้า	ix
รูปที่ 25	แสดงผลลัพท์เมื่อไม่พบถายนิ้วมือที่ค้นหา	ix

คำอธิบายสัญลักษณ์และคำย่อ

E-filing	Electronic filing system ระบบจัดการและค้นหาเอกสารด้วยอิเล็กทรอนิกส์
Pixel	picture element จุดที่เล็กที่สุด รวมตัวกันจนกลายเป็นภาพ
CCD	Charge Coupled Device เป็นเซนเซอร์ที่ทำหน้าที่รับแสงและเปลี่ยนค่าแสงเป็นสัญญาณอนาล็อก และเปลี่ยนสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล
DPI	Dot Per Inch พื้นที่ของจำนวนพิกเซลใน 1 หน่วยตารางนิ้ว

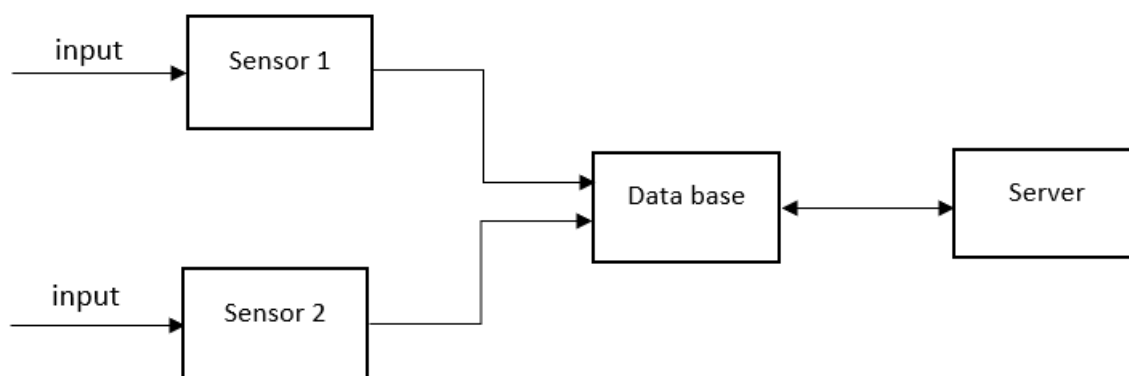
บทที่ 1 บทนำ

1.1 ประวัติความเป็นมา

ในการระบุตัวตนโดยใช้คุณลักษณะทางกายภาพอย่างใดอย่างหนึ่งนั้น เป็นที่นิยมใช้มาอย่างยาวนาน เช่น ลายนิ้วมือ ใบหน้า ม่านตา ดีเอ็นเอ หรือเสียงพูด เมื่อมีการบันทึกคุณลักษณะไว้ก่อนแล้ว จะสามารถระบุตัวบุคคลนั้นได้ การลายนิ้วมือถือเป็นอีกหนึ่งวิธีที่ได้รับความนิยมเป็นอย่างมาก เนื่องจากลายนิ้วมือมีโอกาสเพียง 1 ใน 64,000 ล้าน คนเท่านั้นที่ลายนิ้วมือจะเหมือนกัน ฉะนั้นจึงเป็นคุณลักษณะที่เฉพาะตัวและไม่สามารถเปลี่ยนแปลงตามกาลเวลาได้

ดังนั้นจะเห็นได้ว่าการเช็คชื่อโดยวิธีสแกนลายนิ้วมือนั้นเป็นประโยชน์อย่างมากในการระบุตัวตน เพราะสามารถบันทึกค่าเก็บไว้และแสดงผลผ่านจอแสดงผลได้ เพื่อให้สะดวกต่อการตรวจสอบและเป็นวิธีการที่ไม่สามารถทำแทนกันได้เนื่องจากลายนิ้วมือที่มีลักษณะจำเพาะ

เครื่องสแกนลายนิ้วมือในปัจจุบันนั้นเป็นที่ใช้งานกันอย่างกว้างขวาง จากการศึกษาพบว่าเครื่องสแกนลายนิ้วมือนั้นมีราคาสูง จึงต้องการที่จะจัดทำเครื่องสแกนลายนิ้วที่มีราคาต่ำลง แต่ยังคงรักษาคุณภาพ และทำงานได้เทียบเท่ากับเครื่องสแกนลายนิ้วมือที่วางจำหน่ายทั่วไป นอกจากนี้จะมีการนำเครื่องสแกนลายนิ้วมือนี้นี้ไปใช้งานร่วมกับระบบเช็คชื่อที่สามารถบันทึกค่าลายนิ้วมือและตรวจสอบการเข้าเช็คชื่อได้ โดยมี Block Diagram ดังแสดงในรูปที่ 1.1



รูปที่ 1 แผนผังแสดงการทำงานของระบบ

1.2 วัตถุประสงค์

จากความเป็นมาที่ได้มีการกล่าวถึงข้างต้นนั้น สามารถกำหนดวัตถุประสงค์ของการจัดทำโครงการเพื่อตอบสนองความต้องการของผู้ใช้งานให้มีความสะดวกและรวดเร็วในการใช้งานระบบบันทึกเวลา ดังนี้

1. เพื่อสร้างระบบบันทึกเวลาโดยการสแกนลายนิ้วมือ
2. เพื่อนำ Raspberry Pi มาเป็นหน่วยประมวลผลของระบบ
3. เพื่อสร้างระบบบันทึกเวลาที่สามารถรับและส่งข้อมูลผ่านระบบเครือข่ายไร้สายได้

1.3 ขอบเขต

ขอบเขตการดำเนินงานของโครงการระบบบันทึกเวลาด้วยนิ้วมือ ด้วยบอร์ด Raspberry Pi มีดังนี้

1. ระบบบันทึกเวลาด้วยลายนิ้วมือสามารถบันทึก ID และเวลา ในการเข้าเช็คชื่อได้
2. ระบบบันทึกเวลาด้วยนิ้วมือที่มีการจัดการฐานข้อมูลโดยใช้ SQL
3. ระบบบันทึกเวลาด้วยนิ้วมือที่สามารถรับและส่งข้อมูลผ่านระบบเครือข่ายไร้สายได้
4. ระบบบันทึกเวลาที่สามารถบันทึกข้อมูลได้จากเครื่องสแกนลายนิ้วมือหลายตัวที่มีฐานข้อมูลเดียวกัน

1.4 ผลที่คาดว่าจะได้รับ

ระบบบันทึกเวลาโดยการสแกนลายนิ้วมือ โดยมีคุณสมบัติสามารถระบุตัวตนของเจ้าของลายนิ้วมือนั้นได้ และมีการใช้ Raspberry Pi เป็นหน่วยประมวลผลของระบบ มีการจัดเก็บข้อมูลโดยใช้ SQL ที่สามารถรับและส่งข้อมูลระหว่างผู้ใช้ผ่านระบบเครือข่ายไร้สาย WIFI ที่ติดตั้งมากับบอร์ด Raspberry Pi ได้ นอกจากนี้ยังสามารถบันทึกข้อมูลได้จากเครื่องสแกนลายนิ้วมือหลายตัว ที่มีฐานข้อมูลเดียวกัน เพื่อให้เกิดความสะดวกสบายแก่ผู้ใช้งาน

บทที่ 2 เอกสารที่เกี่ยวข้อง

2.1 Raspberry Pi

Raspberry Pi เปรียบเสมือนคอมพิวเตอร์ขนาดเล็กที่ถูกพัฒนาขึ้นโดย Raspberry Pi Foundation เพื่อให้เป็นคอมพิวเตอร์ที่มีราคาถูกและมีประสิทธิภาพสูง อีกทั้ง Raspberry Pi ยังสามารถเชื่อมต่อกับระบบเครือข่ายแบบใช้สายและแบบไร้สายได้ ทำให้กลายเป็นอุปกรณ์ Internet of Things โดยสมบูรณ์ สามารถนำไปประยุกต์ใช้กับ sensor ในการเก็บข้อมูล และยังสามารถเชื่อมต่อเข้ากับแป้นพิมพ์และเมาส์ได้โดยผ่าน USB port ทำให้ได้รับความนิยมในการนำไปใช้ในการสอนและการทดลองต่าง ๆ

Raspberry Pi นับว่าเป็น Embedded System อีกตัวที่ง่ายต่อการพัฒนา และในปัจจุบัน Raspberry Pi ได้พัฒนาผลิตภัณฑ์มาหลายรุ่นตั้งแต่ Raspberry Pi Model A, Model B, Model B+, Raspberry Pi 2 และ Raspberry Pi 3, Raspberry Pi 4[1]

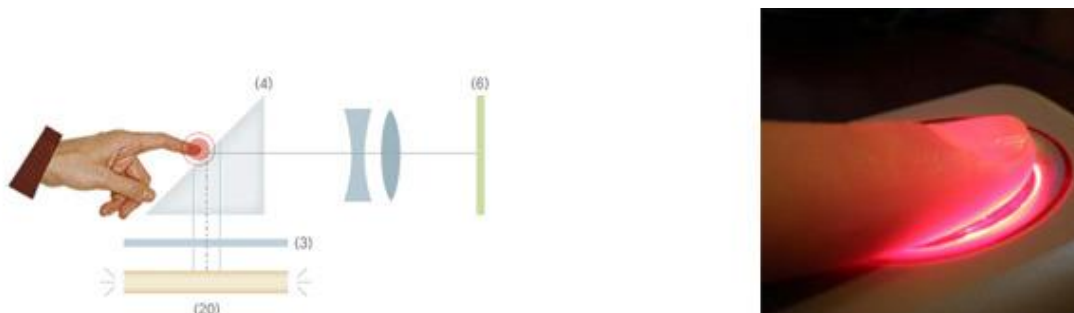


รูปที่ 2 Raspberry Pi

Raspberry Pi รุ่นแรกที่ถูกผลิตออกมา ภายในบอร์ดประกอบไปด้วย 1 USB port, หน่วยความจำหลัก 256เมกะไบต์ มีความเร็วในการประมวลผลที่ 700เมกะเฮิรตซ์ และมีเครือข่ายแบบไร้สายภายในตัวบอร์ด จากนั้นได้มีการพัฒนามาเรื่อย ๆ เพื่อให้เหมาะสมกับการใช้งาน จนในปัจจุบันมีบอร์ด Raspberry Pi ถูกออกแบบให้มี หน่วยประมวลผลที่มีการทำงานถึง 4 คอร์ โดยที่แต่ละคอร์จะทำหน้าที่ที่แตกต่างกันไป นอกจากนี้ยังมีโมดูลสำหรับประมวลผลภาพถ่ายภาพ VideoCore IV และมีหน่วยความจำหลักถึง 512เมกะไบต์ นอกจากนี้ยังประกอบด้วย ไมโคร USB 2.0 HDMI มี

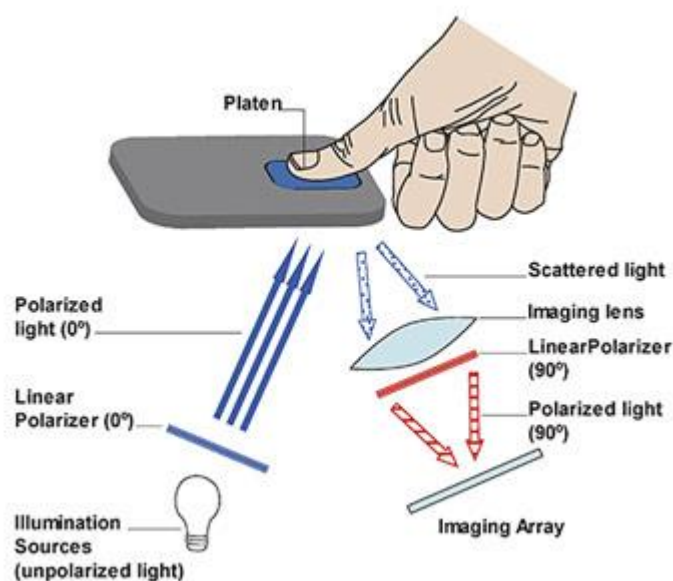
2.2 Fingerprint Sensor

Fingerprint Sensor หรือ เครื่องสแกนลายนิ้วมือ ในปัจจุบันมีอุปกรณ์หลากหลายชนิดที่สามารถนำมาประยุกต์ใช้ในระบบเครื่องสแกนลายนิ้วมือได้ และ sensor ประเภทที่ได้รับความนิยมสูงสุดถึง คือ Optical Sensor



รูปที่ 4 การทำงานของ Optical Sensor

Optical Sensor มีหลักการการสแกนด้วยลำแสงคือ อุปกรณ์ถ่ายเทประจุ เรียกว่า CCD ซึ่งเป็นเซนเซอร์ตัวเดียวกับกล้องถ่ายภาพดิจิทัล และกล้องถ่ายวิดีโอ CCD ทำหน้าที่ในการจัดการข้อมูลแสงบนช่องว่างของแผ่นชั้นส่วนอิเล็กทรอนิกส์ชนิดสองขั้ว(Diodes) ซึ่งสร้างสัญญาณไฟฟ้าเพื่อทำให้เกิดโฟตอน (Photons) ในช่องว่างของแผ่นรับแสงแต่ละช่อง จะบันทึกข้อมูลเป็นพิกเซล เมื่อรวมกันแล้วก็จะเกิดจุดพิกเซลที่เป็นแบบแสงและพิกเซลแบบทึบ จากภาพที่ได้แสงออกมา ซึ่งโดยทั่วไปแล้วเครื่องสแกนนิ้วมือจะทำหน้าที่แปลงสัญญาณอนาล็อก ให้เป็นสัญญาณดิจิทัลภายในตัวของมันเองเนื่องจากใช้เซนเซอร์ CCD การสแกนจะเริ่มขึ้นหลังจากที่เราวางนิ้วของเราไปบนแท่นสแกน และกล้องที่ใช้เซนเซอร์ CCD ก็จะมีการถ่ายภาพ เครื่องสแกนเนอร์ มีแหล่งกำเนิดแสงในตัวมันเองเพื่อทำให้เกิดแสง และมองเห็นเส้นบนที่อยู่นบนนิ้วมือ เซนเซอร์ระบบ CCD จะทำการกลับภาพลายนิ้วมือ เพื่อให้พื้นที่ที่มีความทึบกว่าสะท้อนแสงได้มาก นั่นคือ เส้นบนบนผิวหนังของนิ้ว และให้พื้นที่ที่สว่างกว่า สะท้อนแสงได้น้อยคือส่วนร่องบนผิวหนังของนิ้ว ก่อนที่จะนำภาพที่ได้ไปจัดเก็บ



รูปที่ 5 การทำงานของ Optical Sensor

การบันทึกลายนิ้วมือ เมื่อมีการนำภาพที่ได้ไปจัดเก็บ เครื่องบันทึกลายนิ้วมือก็จะตรวจสอบอีกทีว่า ภาพที่จับได้นั้นมีความชัดเจนมากน้อยเพียงใด จะมีเช็คค่าเฉลี่ยความทึบของพิกเซล และจะปฏิเสธการสแกน หากว่าภาพที่ได้นั้นมีความมืดทึบไป หรือมีความสว่างน้อยไป ถ้าหากมีการปฏิเสธการสแกนเกิดขึ้น เครื่องสแกนจะปรับเวลาการรับแสงให้น้อยลงในกรณีที่ภาพสว่างเกิน และจะมีการขอให้ผู้ใช้งานสแกนใหม่อีกครั้ง ถ้าระดับความเข้มเพียงพอระบบก็จะทำงานต่อ เพื่อจะตรวจสอบความละเอียดของภาพ การประมวลผลจะเริ่มจากการสแกนในแนวนอน และแนวตั้งหลายๆครั้ง ถ้าหากรูปภาพมีความคมชัดเส้นตั้งฉากกับเส้นขนานก็จะถูกสร้างขึ้น เพื่อนำไปสลับกับส่วนที่มีมืดมาก และสว่างมากของภาพ ถ้าหน่วยประมวลผลพบว่า ภาพมีความคมชัดและถูกต้องแล้ว มันก็จะดำเนินการเปรียบเทียบรอยนิ้วมือที่เคยสร้างขึ้นไว้บนไฟล์ในส่วนต่อมา

ข้อดีของเครื่องสแกนนิ้วมือชนิดลำแสง คือ เครื่องมีอายุการใช้งานยาวนาน การบำรุงรักษาง่าย เครื่องมีผลกระทบน้อยจากสภาพแวดล้อมในการใช้งาน และคุณภาพของภาพลายพิมพ์นิ้วมือที่ได้มีความละเอียดในค่าที่ยอมรับได้ในทางสากล คือ 500 dpi หรือมากกว่า ขนาดของตัวเครื่อง fingerscan เอง มีการพัฒนาให้บางและเล็กลงมาก สามารถพัฒนาให้บันทึกนิ้วมือได้ทุกขนาด ถึงแม้ว่า ภาพลายพิมพ์นิ้วมือที่บันทึกได้จาก Sensor ชนิด Optical จะมีความละเอียดสูง แต่ภาพก็มีความบิดเบือนสูงเช่นกันและไม่ได้อัตราส่วนตามขนาดของนิ้วมือจริงในกรณีเช่นนี้สามารถแก้ไขได้โดยเลือกชนิดและวัสดุของปริซึมพร้อมทั้งการจัดวางเรียงตำแหน่งปริซึมและเลนส์ให้เหมาะสมก็จะลดความบิดเบือนของภาพลายพิมพ์นิ้วมือลงไปได้ หรือแก้ไขโดยใช้ซอฟต์แวร์ก็เป็นทางเลือกที่นิยมใช้กันอยู่ในท้องตลาดขณะนี้ นอกจากนี้ความชื้นของพื้นผิวบนนิ้วมือและขนาดลายเส้นนิ้วมือก็ส่งผลถึงการบันทึกภาพด้วย ภาพลายพิมพ์นิ้วมือที่บันทึกได้จะมีความคมชัดและเห็นลายเส้นชัดเจน เมื่อนิ้ว

มือที่ประทับนั้นมีความขึ้นจากเหงื่อเล็กน้อยและลายเส้นค่อนข้างกว้างและลึก ถ้าความขึ้นของนิ้วสูงเกินไป จะได้ภาพลายเส้นหนูนและเส้นลึกติดกันมาก ถ้านิ้วมือแห้งเกินไปก็จะได้ภาพที่เห็นลายเส้น

ที่ไม่ชัดเจน หรือถ้าลายเส้นแคบหรือตื้นเกินไป ภาพลายพิมพ์นิ้วมือที่บันทึกได้ก็จะเป็นวงขนาดเท่ากับนิ้วมือ ซึ่งต้องทำการประทับหรือบันทึกใหม่ อาจจะใช้เวลาในการตรวจในปัจจุบัน มีการนำสารโพลีเมอร์มาติดตั้งบนปริซึมด้านที่ไชรองรับนิ้วมือ เพื่อแก้ไขเรื่องความขึ้นและขนาดลายเส้นของนิ้วมือที่มาสัมผัสกับปริซึมได้ ทำให้สามารถใช้งานได้กับนิ้วมือทุกสภาวะความขึ้นและทุกขนาดได้ดีขึ้น [2]

2.3 ฐานข้อมูล

ฐานข้อมูล หรือ Database คือ กลุ่มของข้อมูลที่ถูกรวบรวมไว้ โดยสัมพันธ์ซึ่งกันและกัน โดยไม่บังคับว่าข้อมูลทั้งหมดนี้จะต้องเก็บไว้ในแฟ้มข้อมูลเดียวกันหรือแยกเก็บหลาย ๆ แฟ้มข้อมูล

ระบบฐานข้อมูล (Database System) คือ ระบบที่รวบรวมข้อมูลต่าง ๆ ที่เกี่ยวข้องกันไว้ด้วยกันอย่างมีระบบ มีความสัมพันธ์ระหว่างข้อมูลต่าง ๆ ที่ชัดเจน ในระบบฐานข้อมูลจะประกอบด้วยแฟ้มข้อมูลหลายแฟ้มที่มีข้อมูลเกี่ยวข้องสัมพันธ์กันเข้าไว้ด้วยกันอย่างเป็นระบบ และเปิดโอกาสให้ผู้ใช้สามารถใช้งานและดูแลรักษาป้องกันข้อมูลเหล่านี้ ได้อย่างมีประสิทธิภาพ โดยมีซอฟต์แวร์ที่เปรียบเสมือนสื่อกลางระหว่างผู้ใช้และโปรแกรมต่าง ๆ ที่เกี่ยวข้องกับการใช้ฐานข้อมูล เรียกว่า ระบบจัดการฐานข้อมูล หรือ DBMS (Data Base Management System) มีหน้าที่ช่วยให้ผู้ใช้เข้าถึงข้อมูลได้ง่ายและมีคุณภาพ การเข้าใช้งานข้อมูลของผู้ใช้อาจเป็นการสร้างฐานข้อมูล การแก้ไขฐานข้อมูล หรือการตั้งคำถามเพื่อให้ได้ข้อมูลมา โดยผู้ใช้ไม่จำเป็นต้องรับรู้เกี่ยวกับรายละเอียดภายในโครงสร้างของฐานข้อมูล

ส่วนประกอบของแฟ้มข้อมูล (File) ระเบียน (Record) และเขตข้อมูล (Field) และถูกจัดการด้วยระบบเดียวกันโปรแกรมคอมพิวเตอร์จะเข้าไปดึงข้อมูลที่ต้องการได้อย่างรวดเร็ว ซึ่งอาจเปรียบฐานข้อมูลเสมือนเป็น Electronic filing system

➤ บิต (bit) ย่อมาจาก Binary Digit ข้อมูลในคอมพิวเตอร์ 1 บิต จะแสดงได้ 2 สถานะคือ 0 หรือ 1 การเก็บข้อมูลต่าง ๆ ได้จะต้องนำบิตหลาย ๆ บิต มาเรียงต่อกัน เช่นนำ 8 บิต มาเรียงเป็น 1 ชุด เรียกว่า 1 ไบต์ (byte)

➤ เขตข้อมูล (field) คือ การนำไบต์ (byte) หลาย ๆ ไบต์ มาเรียงต่อกัน เช่น Name ใช้เก็บชื่อ Last name ใช้เก็บนามสกุล เป็นต้น

➤ ระเบียน (record) คือ การนำเขตข้อมูลหลาย ๆ เขตข้อมูล มาเรียงต่อกัน เช่น แฟ้มข้อมูลนักศึกษา จะเก็บชื่อสกุล รหัสนักศึกษาของนักศึกษาคนที่ 1 เป็นต้น

➤ แฟ้มข้อมูล (File) คือ การเก็บระเบียบหลาย ๆ ระเบียบรวมกัน เช่น แฟ้มเก็บข้อมูลนักศึกษาจำนวน 100 คน เป็นต้น

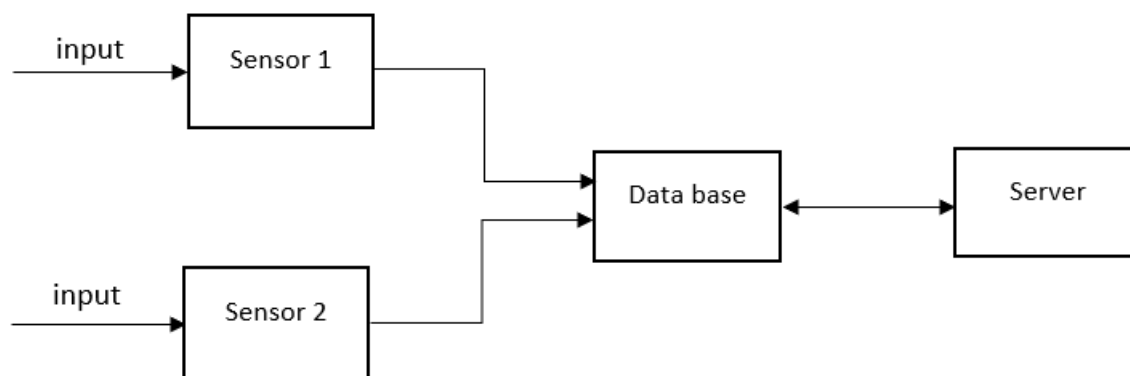
➤ ฐานข้อมูล (database) คือ การจัดเก็บแฟ้มข้อมูลหลาย ๆ แฟ้มข้อมูลไว้ภายใต้ระบบเดียวกัน เช่น เก็บแฟ้มข้อมูล เป็นต้น[3]

2.4 Graphical User Interface

Graphical User Interface หรือ GUI คือ การติดต่อกับผู้ใช้ผ่านการใช้สัญลักษณ์ เป็นการออกแบบส่วนของโปรแกรมคอมพิวเตอร์ให้มีการโต้ตอบกับผู้ใช้ โดยการใช้ Icon รูปภาพ หรือสัญลักษณ์ต่าง ๆ แทนการพิมพ์ชุดคำสั่งในการทำงาน เพื่อช่วยให้ผู้ใช้งานสามารถทำงานได้สะดวก และรวดเร็วขึ้น โดยไม่จำเป็นต้องจดจำคำสั่งต่าง ๆ ของโปรแกรม ถือเป็นวิธีการให้ความสะดวกสบายแก่ผู้ใช้คอมพิวเตอร์ ให้สามารถสื่อสารกับระบบได้ผ่านทางภาพ เช่น ใช้เมาส์กดเลือก Icon แทนการพิมพ์คำสั่งได้ ซึ่งช่วยทำให้เกิดความรวดเร็วในการทำงาน ไม่จำเป็นต้องเสียเวลาในการเรียนรู้และจดจำชุดคำสั่ง ดังที่แสดงในรูปภาพ 6 เป็นการใช้อีคอนในการเปิดใช้งานแอปพลิเคชัน แทนการพิมพ์ชุดคำสั่งในการเรียกใช้งาน ทำให้สะดวกต่อการใช้งานมากขึ้น

บทที่ 3 วิธีการดำเนินการ

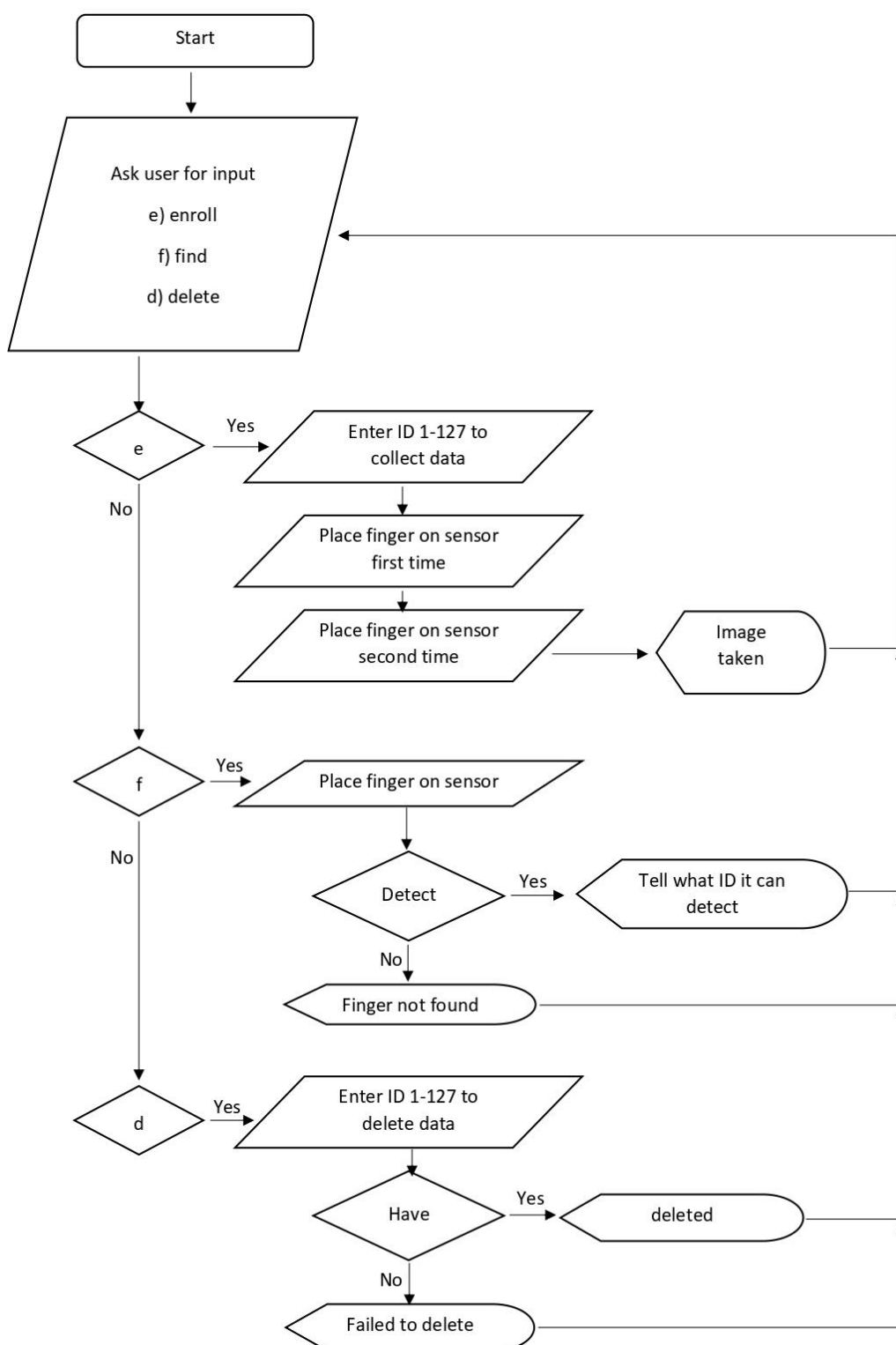
3.1 Block Diagram ของระบบ



รูปที่ 1 แผนผังแสดงการทำงานของระบบ

สำหรับการสร้างเครื่องสแกนลายนิ้ว และเลือกใช้ Sensor DY 50 เป็น Optical sensor สำหรับสแกนลายนิ้วมือ .ในการรับค่า input ซึ่งมีการบันทึก จัดเก็บข้อมูล และตรวจสอบลายนิ้วมือที่ถูกบันทึกค่าไว้ก่อนหน้า ซึ่งภายในเซนเซอร์จะมีหน่วยความจำเก็บบันทึกข้อมูลไว้ จากนั้นจะมีการส่งข้อมูลโดยผ่านเครือข่ายไร้สาย ไปยังฐานข้อมูล โดยภายในโครงการนี้จะใช้คำสั่ง SQL ในการรับข้อมูลลายนิ้วมือจากเซนเซอร์และใช้เป็นฐานข้อมูลจัดเก็บและส่งข้อมูลลายนิ้วมือผ่านทางเครือข่ายไร้สายไปยังผู้ใช้งาน ส่วนของโปรแกรมที่ใช้ในการควบคุมการทำงานของเซนเซอร์นั้นเลือกใช้ Python ในการควบคุมเนื่องจากมีการใช้งานอย่างแพร่หลายและได้รับความนิยมเป็นอย่างมากในปัจจุบัน

3.2 หลักการทำงานของโปรแกรมไพธอน



รูปที่ 6 แผนผังแสดงการทำงานของโปรแกรมไพธอน

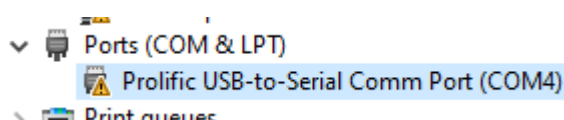
จากรูปที่ 7 แสดงแผนผังการทำงานของโปรแกรมไพธอนที่ใช้ควบคุมการทำงานของ fingerprint โปรแกรมสามารถควบคุมให้บันทึกลายนิ้วมือได้จากการใช้คำสั่ง e โดยสามารถบันทึกได้ทั้งหมด 127 Address ในการบันทึกนั้นจะสแกนนิ้วมือสองรอบเพื่อตรวจสอบความถูกต้อง จากนั้นลายนิ้วมือที่สแกนจะถูกเก็บบันทึกไว้ในหน่วยความจำของเซนเซอร์ และสามารถค้นหาลายนิ้วมือที่ถูกบันทึกไว้ก่อนหน้านี้ได้ด้วยการใช้คำสั่ง f เมื่อไม่พบลายนิ้วมือที่ต้องการค้นหาโปรแกรมจะแสดงผล finger not found นอกจากนี้ภายในโปรแกรมยังมีคำสั่งสำหรับลบลายนิ้วมือที่ถูกบันทึกไว้ก่อนหน้านี้ได้ โดยใช้คำสั่ง d ในการลบข้อมูลลายนิ้วมือที่ถูกบันทึกไว้ก่อนหน้านี้

3.3 วิธีการทดลอง

3.3.1. การเชื่อมต่อ USB to Serial port



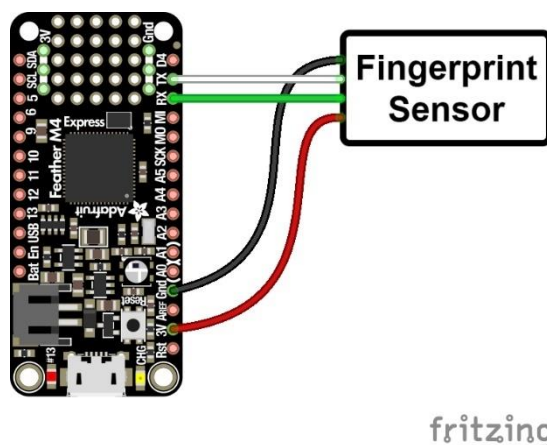
รูปที่ 7 USB to Serial port



รูปที่ 8 แสดงผล windows ไม่สามารถเชื่อมต่อกับ USB ได้

ในการเชื่อมต่อ Fingerprint เข้ากับ windows นั้น จะเชื่อมต่อผ่านอุปกรณ์ USB ดังรูปที่ 8 จะพบว่าที่บอร์ดมีไอซี CH340 ซึ่งเป็นโมดูลที่ใช้แปลง USB เป็น Serial เมื่อเชื่อมต่อ USB เข้ากับคอมพิวเตอร์แล้ว และจากรูปที่ 9 จะพบว่า USB ไม่สามารถเชื่อมต่อเข้ากับ windows ได้ จึงต้องติดตั้ง CH340G Driver for Windows ก่อน เพื่อแปลง USB เป็น Serial[4]

การเชื่อมต่อเซนเซอร์สแกนนิ้วมือ DY50 เข้ากับบอร์ด USB สามารถเชื่อมต่อสายไปยัง Sensor ดังนี้ Sensor VCC ต่อเข้ากับ USB 3.3 V, Sensor GND ต่อเข้ากับ USB Ground, Sensor RX ต่อเข้ากับ USB TX, Sensor TX ต่อเข้ากับ USB RX ดังที่แสดงในรูปที่ 10



รูปที่ 9 การต่อสายจากเซนเซอร์เข้า USB

3.3.2 การใช้โปรแกรมภาษาไพธอน

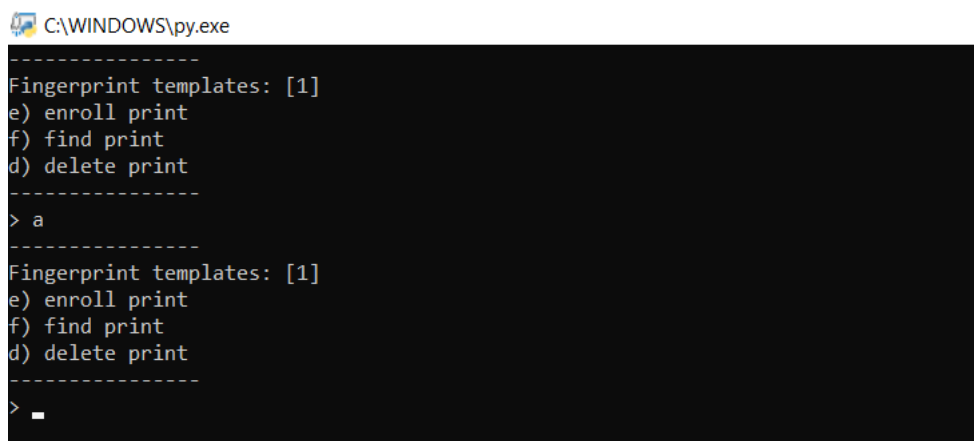
โครงการฉบับนี้เลือกใช้โปรแกรมคำสั่งภาษาไพธอนในการสร้างคำสั่งเพื่อควบคุมการทำงานของ Fingerprint โดยใช้ Library Adafruit CircuitPython Fingerprint โดยประกอบด้วยคำสั่ง 3 คำสั่ง คือ คำสั่งสำหรับลงทะเบียนลายนิ้วมือที่ยังไม่เคยทำการบันทึกมาก่อนหน้า โดยใช้ฟังก์ชัน `enroll_finger` ในการลงทะเบียนลายนิ้วมือ ทำการวางนิ้วที่ต้องการบันทึกลงบนเซนเซอร์สแกนลายนิ้วมือทั้งหมด 2 ครั้ง หากลายมือทั้งสองครั้งนั้นไม่สัมพันธ์กัน จะไม่สามารถบันทึกลายนิ้วมือได้ นอกจากนี้ในคำสั่งการลงทะเบียนลายนิ้วมือยังเพิ่มฟังก์ชันให้สามารถบันทึกข้อมูลส่วนตัวเบื้องต้นของเจ้าของลายนิ้วมือนั้น ๆ ได้ คำสั่งต่อมาคือคำสั่งสำหรับค้นหาลายนิ้วมือที่เคยบันทึกได้ก่อนหน้านี้แล้ว โดยใช้ฟังก์ชัน `get_fingerprint` ซึ่งลายนิ้วที่ได้มีการบันทึกจะถูกเก็บไว้ในหน่วยความจำ และเมื่อตรวจสอบพบลายนิ้วมือที่เคยบันทึกไว้ จะมีการแสดงผลข้อมูลเบื้องต้นของเจ้าของลายนิ้วมือนั้นด้วย และคำสั่งสุดท้ายจะเป็นคำสั่งสำหรับลบลายนิ้วมือที่เคยบันทึกไว้ก่อนหน้านี้ โดยฟังก์ชัน `finger.delete_model` โปรแกรมจะให้กรอกรหัสนักศึกษาทุกครั้งก่อนทำการลบลายนิ้วมือ

3.3.3 สร้างฐานข้อมูล

มีการสร้างไฟล์เอกสาร csv เป็นไฟล์สำหรับเก็บข้อมูลรหัสนักศึกษาของเจ้าของลายนิ้วมือ และเลือกใช้ Library pandas และใช้คำสั่ง `read_csv` ในการอ่านไฟล์ csv ใช้คำสั่งโปรแกรมภาษาไพธอนในการเพิ่มรหัสนักศึกษาเข้าไปในไฟล์ และลบออกจากไฟล์

บทที่ 4 การประเมินผล

4.1 การลงทะเบียนลายนิ้วมือ



```
C:\WINDOWS\py.exe
-----
Fingerprint templates: [1]
e) enroll print
f) find print
d) delete print
-----
> a
-----
Fingerprint templates: [1]
e) enroll print
f) find print
d) delete print
-----
> _
```

รูปที่ 10 แสดงผลหน้าโปรแกรม

จากรูปที่ 11 จะแสดงหน้าจอของโปรแกรม ซึ่งจะมีคำสั่งทั้งหมด 3 คำสั่งคือ คำสั่ง e ใช้สำหรับบันทึกลายนิ้วมือ คำสั่ง f ใช้สำหรับค้นหาลายนิ้วมือที่บันทึกไว้ก่อนหน้านี้ และคำสั่ง d ใช้สำหรับลบลายนิ้วมือที่ได้บันทึกไว้แล้ว เมื่อเลือกใช้คำสั่งนอกเหนือจากคำสั่ง e, f, และ d โปรแกรมจะไม่ทำงานดังรูปที่ 11 และใช้ฟังก์ชันในโปรแกรมภาษาไพธอนในการสร้างคำสั่งสำหรับเลือกการทำงานของ Fingerprint ดังรูปที่ 12

```
if c == "e":
    enroll_finger(get_num())
if c == "f":
    if get_fingerprint():
        print("Detected #", finger.finger_id, "with confidence", finger.confidence)
    else:
        print("Finger not found")
if c == "d":
    if finger.delete_model(get_num()) == adafruit_fingerprint.OK:
        print("Deleted!")
    else:
        print("Failed to delete")|
```

รูปที่ 11 คำสั่งโปรแกรมไพธอนสำหรับเลือกโหมดการทำงานของโปรแกรม

```

-----
Fingerprint templates: [1]
e) enroll print
f) find print
d) delete print
-----
> e
Enter ID # from 1-127: 2_

```

รูปที่ 12 แสดงผลเมื่อเรียกใช้คำสั่ง e

เมื่อใช้คำสั่ง e สำหรับการลงทะเบียนลายนิ้วมือ โปรแกรมจะให้พิมพ์ ID Address เพื่อระบุตำแหน่งที่อยู่ในการเก็บลายนิ้วมือ โดยสามารถใช้ได้ตั้งแต่หมายเลข 1 ถึง 127 ดังแสดงในรูปที่ 12

```

Fingerprint templates: [1]
e) enroll print
f) find print
d) delete print
-----
> e
Enter ID # from 1-127: 2
Place finger on sensor.....
.....Image taken
Templating...Templated
Remove finger
Place same finger again.....Image taken
Templating...Templated
Creating model...Created
Storing model #2...Stored
-----
Storing model #1...Stored
ID: 613040320-5

```

รูปที่ 13 แสดงผลเมื่อบันทึกลายนิ้วมือสำเร็จ

หลังจากพิมพ์ ID Address เสร็จสิ้น ใช้นิ้วมือนิ้วที่ต้องการบันทึกและไปที่เซนเซอร์เพื่อทำการอ่านลายนิ้วมือ โดยในการบันทึกลายนิ้วมือแต่ละครั้งนั้น จะต้องแตะนิ้วมือที่เซนเซอร์จำนวนสองครั้งเพื่อตรวจสอบความถูกต้อง ในการสแกนครั้งแรกจะเก็บไว้เพื่อเป็นแม่แบบในการตรวจสอบความถูกต้อง โดยที่เมื่อมีการเก็บลายนิ้วมือเรียบร้อยแล้ว โปรแกรมจะแสดงผล Templated จากนั้นให้เอานิ้วมือออกจากเซนเซอร์จนกว่าโปรแกรมจะแสดงผล Place same finger again จึงทำการสแกนนิ้วมือรอบต่อไป เมื่อการบันทึกลายนิ้วมือสำเร็จ โปรแกรมจะแสดงผล Creating model...Created และจะแสดงผล Storing model เป็นที่จัดเก็บลายนิ้วมื่อดังกล่าว ดังที่แสดงในรูปที่ 13 ซึ่งจากรูปจะเป็นการเก็บลายนิ้วมือไว้ที่ตำแหน่งที่ 2 เมื่อโปรแกรมแสดงผล Stored จากนั้นโปรแกรมจะให้บันทึกรหัสนักศึกษาของเจ้าของลายนิ้วมือลงไป นั่นหมายถึงการบันทึกลายนิ้วมือเสร็จสมบูรณ์

โดยรูปที่ 15 จะแสดงคำสั่งของโปรแกรมไพธอนที่ควบคุมการจัดเก็บลายนิ้วมือ มีการใช้ฟังก์ชัน `enroll_finger` ในการลงทะเบียนลายนิ้วมือ เมื่อมีเก็บลายนิ้วมือเกิดขึ้น จะใช้คำสั่ง `adafruit.get_image()` ถ้าการสแกนลายนิ้วมือไม่พบปัญหา โปรแกรมจะแสดงผล `Image taken` นั่นคือ ลายนิ้วมือได้ถูกจัดเก็บไว้แล้ว

```
def get_fingerprint_detail():
    """Get a finger print image, template it, and see if it matches!
    This time, print out each error instead of just returning on failure"""
    print("Getting image...", end="", flush=True)
    i = finger.get_image()
    if i == adafruit_fingerprint.OK:
        print("Image taken")
    else:
        if i == adafruit_fingerprint.NO_FINGER:
            print("No finger detected")
        elif i == adafruit_fingerprint.IMAGE_FAIL:
            print("Imaging error")
        else:
            print("Other error")
    return False
```

รูปที่ 14 คำสั่งโปรแกรมไพธอนสำหรับเก็บลายนิ้วมือ

4.2 การค้นหาลายนิ้วที่ลงทะเบียนไว้

```
> f
Waiting for image...
Templating...
Searching...
Detected # 1 with confidence 92
  Number      ID
0         1  613040320-5
-----
```

รูปที่ 15 แสดงผลเมื่อเรียกใช้คำสั่ง `f` ในการค้นหาลายนิ้วมือที่บันทึกไว้ก่อนหน้านี้

```
def get_fingerprint():
    """Get a finger print image, template it, and see if it
    matches!"""
    print("Waiting for image...")
    while finger.get_image() != adafruit_fingerprint.OK:
        pass
    print("Templating...")
    if finger.image_2_tz(1) != adafruit_fingerprint.OK:
        return False
    print("Searching...")
    if finger.finger_search() != adafruit_fingerprint.OK:
        return False
    return True
```

รูปที่ 16 คำสั่งโปรแกรมไพธอนสำหรับค้นหาลายนิ้วมือ

จากรูปที่ 16 พบว่าเมื่อใช้คำสั่ง f ในการค้นหาลายนิ้วมือที่ได้มีการลงทะเบียนไว้ก่อนหน้านี้ ใช้ลายนิ้วมือที่ได้มีการบันทึกไว้และไปที่เซนเซอร์เพื่อค้นหาลายนิ้วมือนั้น โปรแกรมจะแสดงผล Detected # ID Address เมื่อลายนิ้วมือที่ทำการค้นหาตรงกับที่เคยลงทะเบียนไว้ โปรแกรมจะแสดงผลที่อยู่ของลายนิ้วมือนั้นและแสดงรหัสนักศึกษาที่ได้มีการบันทึกไว้ไว้ สำหรับคำสั่งโปรแกรมภาษาไพธอนในการค้นหาลายนิ้วมือนั้น ใช้ฟังก์ชัน get_fingerprint ดังรูปที่ 17

```
-----
Fingerprint templates: [1, 2]
e) enroll print
f) find print
d) delete print
-----
> f
Waiting for image...
Templating...
Searching...
Finger not found
-----
```

รูปที่ 17 แสดงผลเมื่อใช้คำสั่ง f ค้นหาลายนิ้วมือไม่สำเร็จ

เมื่อลายนิ้วมือที่ทำการค้นหานั้นไม่ตรงกับลายนิ้วมือที่ได้มีการลงทะเบียนไว้ก่อนหน้านี้ โปรแกรมจะแสดงผล Finger not found ดังรูปที่ 18

4.3 การลบลายนิ้วมือ

```
-----
Fingerprint templates: [1, 2]
Templat
e) enroll print
f) find print
d) delete print
-----
> d
Enter ID # from 1-127: 1
Please enter a ID to be deleted:613040320-5
```

รูปที่ 18 แสดงผลเมื่อใช้คำสั่ง d ในการลบลายนิ้วมือ

จากรูปที่ 19 เป็นการใช้คำสั่ง d ในการลบลายนิ้วมือที่ได้บันทึกไว้ก่อนหน้านี้ ทำได้โดยการพิมพ์ ID Address ที่จัดเก็บลายนิ้วมือนั้นไว้ โดยมีที่อยู่ตั้งแต่ 1-127 จากนั้นพิมพ์รหัสนักศึกษาที่ต้องการลบเพื่อให้รหัสนักศึกษาถูกลบจากไฟล์ฐานข้อมูล จากนั้นโปรแกรมจะแสดง Deleted! เมื่อการลบลายนิ้วมือนั้นสำเร็จ และจะแสดงผล Failed to delete เมื่อการลบไม่สำเร็จ

4.4 การใช้ฐานข้อมูล

	A	B	C	D	E	F	G
1	Number	ID					
2	1	613040320-5					
3	2	613040320-6					
4							
5							
6							
7							
8							
9							

รูปที่ 19 ไฟล์ข้อมูล .csv ที่เก็บข้อมูลรหัสนักศึกษา

เมื่อมีการลงทะเบียนรายใหม่นี้ จากหัวข้อ 4.1 โปรแกรมจะให้ใส่รหัสนักศึกษาด้วยทุกครั้ง หลังจากตรวจสอบรายใหม่นี้เสร็จสิ้น แล้วรหัสนักศึกษาที่มีการลงทะเบียนไว้จะเข้ามาอยู่ไฟล์ csv ที่สร้างไว้โดยใช้คำสั่ง `s.append(id)` ดังรูปที่ 21 และเมื่อทำการลบรายใหม่นี้ ออก ดังหัวข้อ 4.3 จะทำให้รหัสนักศึกษาที่อยู่ในไฟล์ถูกลบออกไปด้วย

```
def data():
    import pandas as pd

    id = input(str("ID: "))
    s.append(id)

    dflist = pd.DataFrame(list(zip(r, s)),
                          columns=["Number", "ID"])
    dflist.to_csv("iddata.csv", index = False)
```

รูปที่ 20 คำสั่งโปรแกรมไพธอนสำหรับเพิ่มรหัสนักศึกษาในไฟล์ข้อมูล

```
def read():
    import csv
    import pandas as pd
    df = pd.read_csv("iddata.csv")
    dflist = pd.DataFrame(df)

    print(dflist[dflist["Number"] == finger.finger_id])

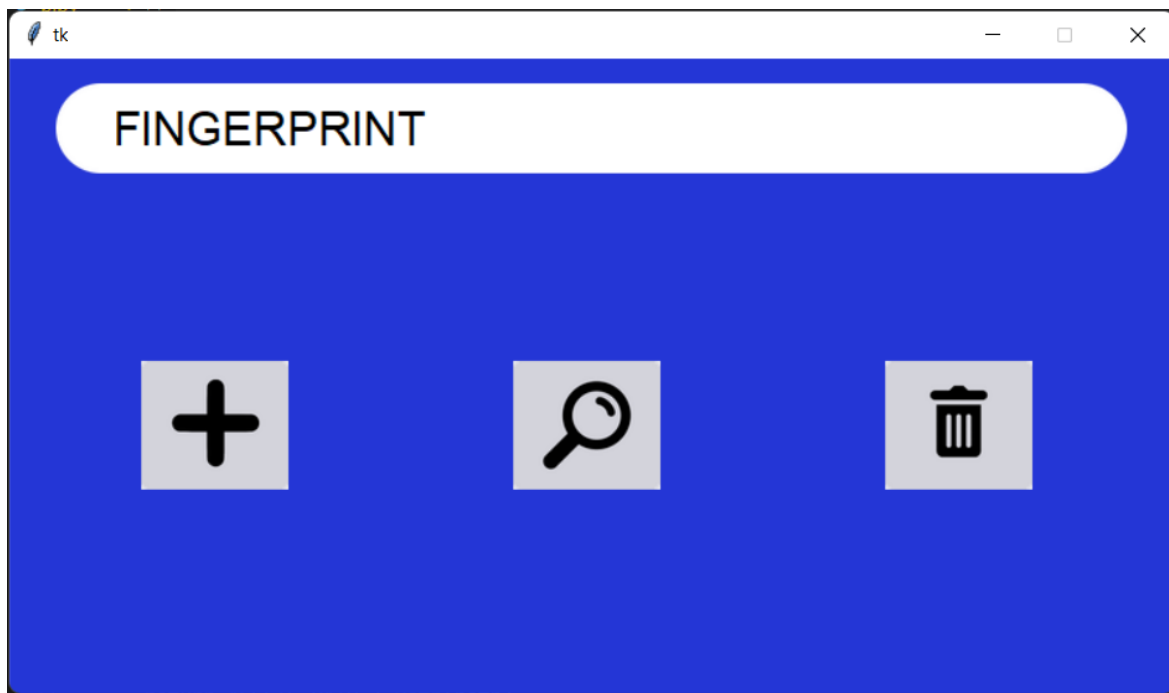
def deletespec():
    import csv
    import pandas as pd
    members= input("Please enter a ID to be deleted:")
    df = pd.read_csv("iddata.csv")
    dflist = pd.DataFrame(df)
    #drin = dflist[dflist["ID"] == members]




    #dflist = pd.DataFrame(list(zip(r, s)),
    #                      #columns=["Number", "ID"])

    dflist.drop(dflist.index[dflist['ID'] == members], inplace = True)
```

รูปที่ 21 คำสั่งโปรแกรมไพธอนสำหรับอ่านและลบรหัสนักศึกษาจากไฟล์ข้อมูล

4.5 Graphical User Interface



การสร้าง GUI เพื่อความสะดวกในการใช้งาน เป็นการกดปุ่มสัญลักษณ์แทนการพิมพ์คำสั่งการทำงาน ภายในโครงการนี้จะกำหนดให้สัญลักษณ์  แทนการใช้คำสั่ง e เพื่อการบันทึกลายนิ้วมือใหม่ ใช้สัญลักษณ์  แทนคำสั่ง f ในการค้นหาลายนิ้วมือที่มีการเก็บบันทึกไว้ก่อนหน้านี้ และใช้สัญลักษณ์  แทนคำสั่ง d เป็นการลบลายนิ้วมือที่มีการบันทึกไว้แล้ว

บทที่ 5 สรุปและอภิปราย

ระบบบันทึกเวลาด้วยลายนิ้วมือ โดยใช้ Raspberry Pi เป็นหน่วยประมวลผลของระบบ โดยในโครงการฉบับนี้เบื้องต้นได้ทดลองใช้โปรแกรมภาษาไพธอนเป็นคำสั่งในการควบคุมเครื่องสแกนลายนิ้วมือ โดยภายในโปรแกรมมีคำสั่งเบื้องต้นเพื่อให้เครื่องสแกนลายนิ้วมือสามารถบันทึกลายนิ้วมือผู้ใช้ สามารถค้นหาลายนิ้วมือที่ถูกบันทึกไว้ก่อนหน้านี้ และมีคำสั่งลบลายนิ้วมือที่มีการลงทะเบียนไว้แล้ว ข้อมูลลายนิ้วมือที่จัดเก็บไว้ในหน่วยความจำของเซนเซอร์นั้น จะถูกส่งไปยังฐานข้อมูล MySQL โดยการส่งข้อมูลผ่านเครือข่ายไร้สาย มีการใช้ TKinter ในการสร้าง Graphical User Interface (GUI) เพื่อให้โปรแกรมสามารถมีความสะดวกสบายและรวดเร็วในการใช้งาน

จากการทดลองพบว่า โปรแกรมภาษาไพธอนที่เขียนขึ้นเพื่อควบคุมคำสั่งเครื่องสแกนลายนิ้วมือนั้น สามารถบันทึกลายนิ้วมือได้สูงสุด 127 ผู้ใช้ และสามารถค้นหาลายนิ้วมือที่เคยมีการบันทึกไว้ก่อนหน้านี้ได้ จากการทดลองใช้งานเบื้องต้นไม่พบการทำงานที่ผิดพลาด สามารถค้นหาลายนิ้วมือที่เคยบันทึกไว้ได้อย่างแม่นยำ

แนวทางการพัฒนาระบบในอนาคต พัฒนาระบบบันทึกเวลาที่ใช้ Raspberry Pi ในการประมวลผลของระบบ มีการส่งข้อมูลการบันทึกลายนิ้วมือไปยังเซิร์ฟเวอร์ที่ให้บริการ โดยส่งผ่านระบบเครือข่ายไร้สายที่มีอยู่ภายในบอร์ด Raspberry Pi

เอกสารอ้างอิง

- [1] Worachet M., “BASIC RASPBERRY PI For Internet of Things.”
- [2] บริษัท ฟิงเทค จำกัด, “Sensor.” fingerscan.in.th/fingerprint-scanner/51-optical-scanner-finger-scan (accessed Oct. 14, 2021).
- [3] พลชัย พิทักษานนท์กุล, “database.” glurgeek.com/education/ระบบฐานข้อมูล-database-system/ (accessed Oct. 14, 2021).
- [4] E. Datasheet, “The DataSheet of CH340 (the first) USB to serial chip CH340.” [Online]. Available: <http://wch.cn>

ภาคผนวก

ภาคผนวก ก โปรแกรมภาษาไพธอน

```

import time
from numpy.testing._private.utils import jiffies
import serial
import adafruit_fingerprint
uart = serial.Serial(port="COM6", baudrate=115200, timeout = 1)
finger = adafruit_fingerprint.Adafruit_Fingerprint(uart)

def get_fingerprint():
    """Get a finger print image, template it, and see if it
    matches!"""
    print("Waiting for image...")
    while finger.get_image() != adafruit_fingerprint.OK:
        pass
    print("Templating...")
    if finger.image_2_tz(1) != adafruit_fingerprint.OK:
        return False
    print("Searching...")
    if finger.finger_search() != adafruit_fingerprint.OK:
        return False
    return True

# pylint: disable=too-many-branches
def get_fingerprint_detail():
    """Get a finger print image, template it, and see if it matches!
    This time, print out each error instead of just returning on
    failure"""
    print("Getting image...", end="", flush=True)
    i = finger.get_image()
    if i == adafruit_fingerprint.OK:
        print("Image taken")
    else:
        if i == adafruit_fingerprint.NO_FINGER:
            print("No finger detected")
        elif i == adafruit_fingerprint.IMAGE_FAIL:
            print("Imaging error")
        else:
            print("Other error")
    return False

    print("Templating...", end="", flush=True)
    i = finger.image_2_tz(1)
    if i == adafruit_fingerprint.OK:
        print("Templated")
    else:
        if i == adafruit_fingerprint.IMAGE_MESS:

```



```

        print("Image too messy")
    elif i == adafruit_fingerprint.FEATUREFAIL:
        print("Could not identify features")
    elif i == adafruit_fingerprint.INVALIDIMAGE:
        print("Image invalid")
    else:
        print("Other error")
    return False

print("Searching...", end="", flush=True)
i = finger.finger_fast_search()
# pylint: disable=no-else-return
# This block needs to be refactored when it can be tested.
if i == adafruit_fingerprint.OK:
    print("Found fingerprint!")
    return True
else:
    if i == adafruit_fingerprint.NOTFOUND:
        print("No match found")
    else:
        print("Other error")
    return False

# pylint: disable=too-many-statements
def enroll_finger(location):
    """Take a 2 finger images and template it, then store in
    'location'"""
    for fingerimg in range(1, 3):
        if fingerimg == 1:
            print("Place finger on sensor...", end="", flush=True)
        else:
            print("Place same finger again...", end="", flush=True)

        while True:
            i = finger.get_image()
            if i == adafruit_fingerprint.OK:
                print("Image taken")
                break
            if i == adafruit_fingerprint.NOFINGER:
                print(".", end="", flush=True)
            elif i == adafruit_fingerprint.IMAGEFAIL:
                print("Imaging error")
                return False
            else:
                print("Other error")
                return False

    print("Templating...", end="", flush=True)

```

```

i = finger.image_2_tz(fingerimg)
if i == adafruit_fingerprint.OK:
    print("Templated")
else:
    if i == adafruit_fingerprint.IMAGEMESS:
        print("Image too messy")
    elif i == adafruit_fingerprint.FEATUREFAIL:
        print("Could not identify features")
    elif i == adafruit_fingerprint.INVALIDIMAGE:
        print("Image invalid")
    else:
        print("Other error")
    return False

if fingerimg == 1:
    print("Remove finger")
    time.sleep(1)
    while i != adafruit_fingerprint.NOFINGER:
        i = finger.get_image()

print("Creating model...", end="", flush=True)
i = finger.create_model()
if i == adafruit_fingerprint.OK:
    print("Created")
else:
    if i == adafruit_fingerprint.ENROLLMISMATCH:
        print("Prints did not match")
    else:
        print("Other error")
    return False

print("Storing model #d..." % location, end="", flush=True)
i = finger.store_model(location)
if i == adafruit_fingerprint.OK:
    print("Stored")
else:
    if i == adafruit_fingerprint.BADLOCATION:
        print("Bad storage location")
    elif i == adafruit_fingerprint.FLASHERR:
        print("Flash storage error")
    else:
        print("Other error")
    return False

return True

```

```
#####
r = []
s = []

def data():

    import pandas as pd

    id = input(str("ID: "))
    s.append(id)

    dflist = pd.DataFrame(list(zip(r, s)),
        columns=["Number","ID"])
    dflist.to_csv("iddata.csv", index = False)
    #dic = {"Number": r., "ID": id}
    #dataframe = pd.DataFrame(dic)
    #dataframe.to_csv('iddata.csv')
    #with open("iddata.csv", "a", newline='') as f:

        #writer = csv.DictWriter(f, fieldnames=header)
        #writer.writerow(dic)

def read():
    import csv
    import pandas as pd
    df = pd.read_csv("iddata.csv")
    dflist = pd.DataFrame(df)

    print(dflist[dflist["Number"] == finger.finger_id])

def deletespec():
    import csv
    import pandas as pd
    members= input("Please enter a ID to be deleted:")
    df = pd.read_csv("iddata.csv")
    dflist = pd.DataFrame(df)
    #drin = dflist[dflist["ID"] == members]

    #dflist = pd.DataFrame(list(zip(r, s)),
        #columns=["Number","ID"])

    dflist.drop(dflist.index[dflist['ID'] == members], inplace = True)
    #dflist.set_index("ID")
    #dflist.drop(index = members, inplace=True)
    #dflist.set_index("Number")

    #data.drop(members, axis=0)
```

```

dflist.to_csv("iddata.csv", index = False)

def get_num():
    """Use input() to get a valid number from 1 to 127. Retry till
    success!"""
    i = 0
    while (i > 127) or (i < 1):
        try:
            i = int(input("Enter ID # from 1-127: "))
            r.append(i)
        except ValueError:
            pass
    return i

while True:
    print("-----")
    if finger.read_templates() != adafruit_fingerprint.OK:
        raise RuntimeError("Failed to read templates")
    print("Fingerprint templates:", finger.templates)
    print("e) enroll print")
    print("f) find print")
    print("d) delete print")
    print("-----")
    c = input("> ")

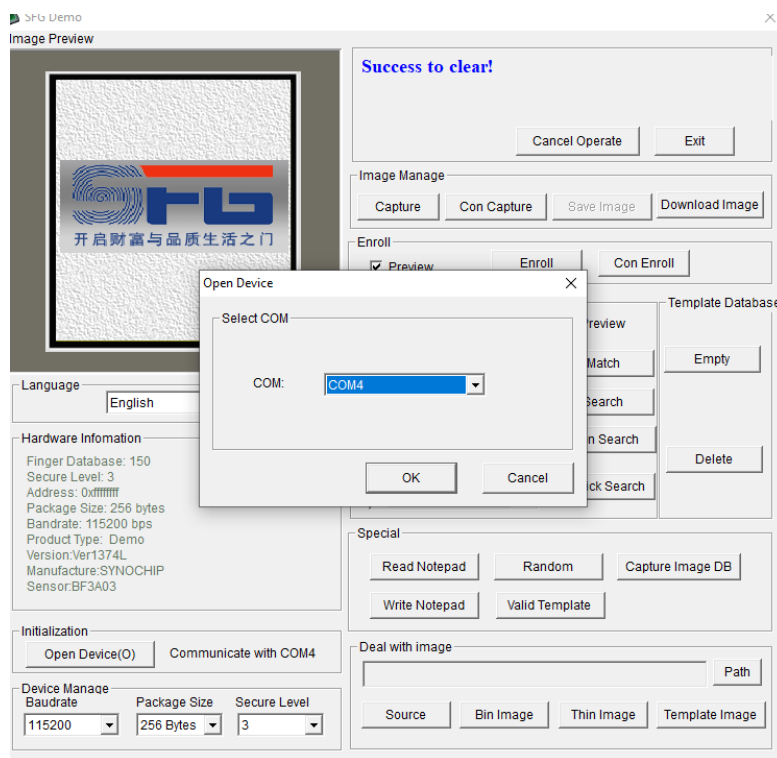
    if c == "e":
        enroll_finger(get_num())
        data()
    if c == "f":
        if get_fingerprint():
            print("Detected #", finger.finger_id, "with confidence",
finger.confidence)
            read()
        else:
            print("Finger not found")
    if c == "d":
        if finger.delete_model(get_num()) == adafruit_fingerprint.OK:
            deletespec()
            print("Deleted!")
        else:
            print("Failed to delete")

```

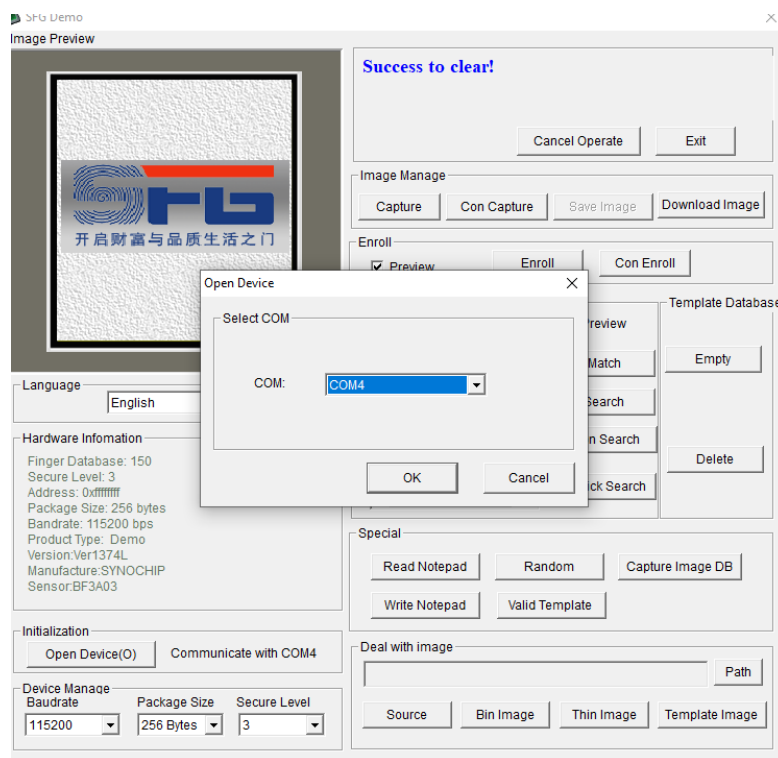
ภาคผนวก ข

ภาคผนวก ข ใช้งาน Fingerprint ร่วมกับโปรแกรม SFG Demo V.2.0

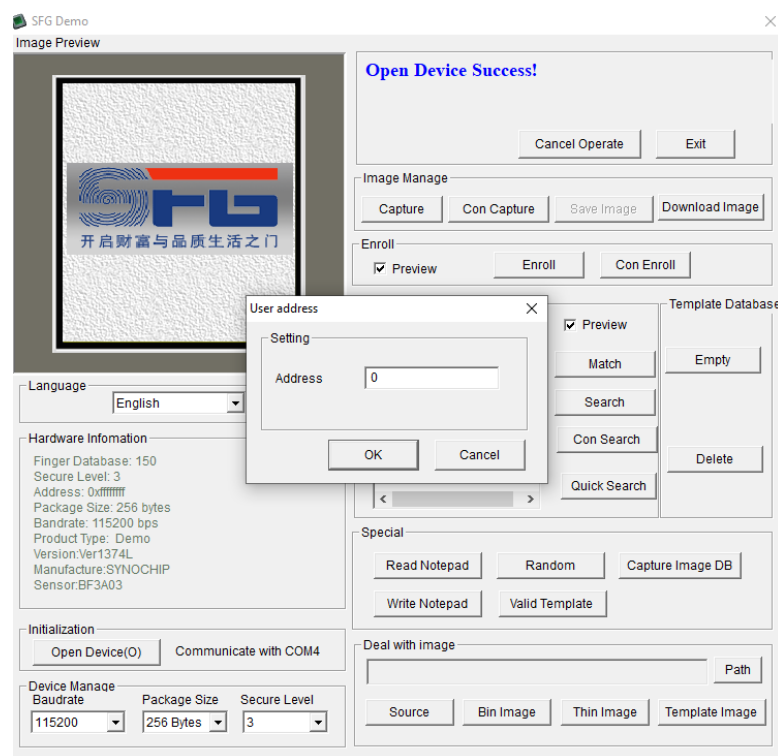
โปรแกรม SFG Demo V.2.0 เป็นโปรแกรมสำเร็จรูป ใช้สำหรับทดสอบการทำงานของ fingerprint ภายในโปรแกรมสามารถบันทึกลายนิ้วมือ ค้นหาลายนิ้วมือที่บันทึกไว้ก่อนหน้านี้ และสามารถลบลายนิ้วมือที่เคยบันทึกไว้แล้ว เมื่อเชื่อมต่อ Fingerprint เข้ากับโปรแกรมแล้ว ทำการเลือกช่องการเชื่อมต่อของอุปกรณ์ ดังแสดงในรูปที่ 4.3 จากนั้นกำหนด Address ของลายนิ้วมือที่ต้องการบันทึกโดยการเลือกใช้คำสั่ง Enroll ดังแสดงในรูปที่ 4.4 โดยในการบันทึกลายนิ้วมือนั้น ระบบจะบังคับให้แตะนิ้วมือไปที่เครื่องสแกนจำนวนสองครั้ง เพื่อตรวจสอบความถูกต้องจึงจะทำการบันทึกได้สำเร็จ จากนั้นใช้คำสั่ง Match ในการค้นหาลายนิ้วมือที่บันทึกไว้ก่อนหน้านี้ เมื่อลายนิ้วมือที่ต้องการค้นหาตรงกับ Address ที่บันทึกไว้แล้ว โปรแกรมจะแสดงผล Pass ดังแสดงในรูปที่ 4.5 และจะแสดงผล Deny ตามรูปที่ 4.6 เมื่อการค้นหานั้นเกิดความผิดพลาด



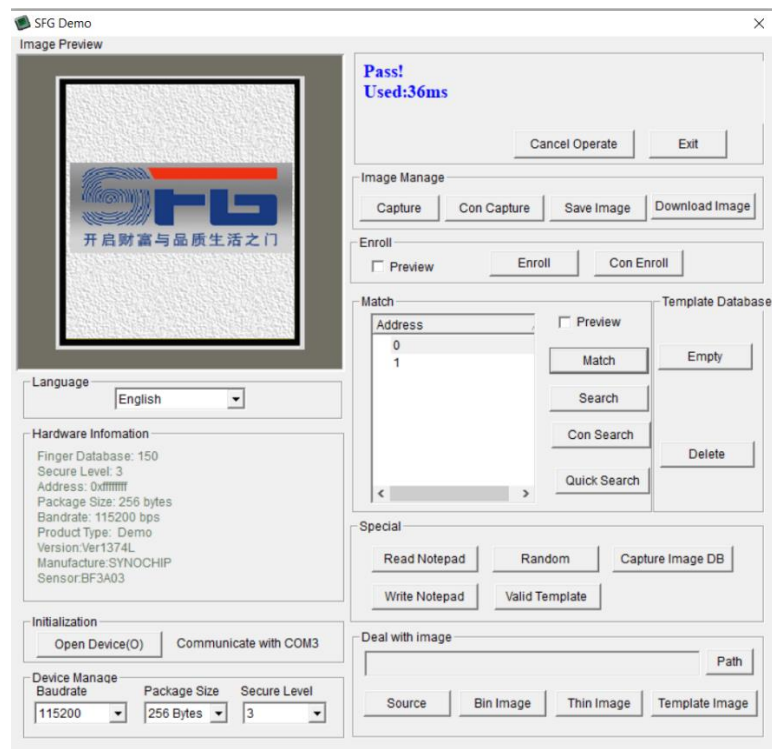
รูปที่ 22 เลือก Port สำหรับเชื่อมต่อกับ Fingerprint



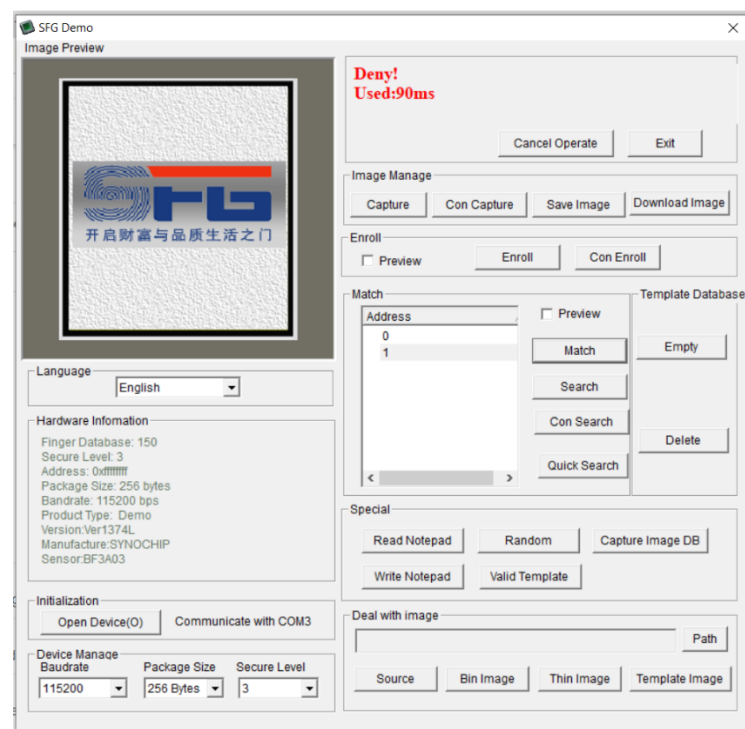
รูปที่ 23 เลือก Port สำหรับเชื่อมต่อกับ Fingerprint



รูปที่ 24 กำหนด Address ของลายนิ้วมือ



รูปที่ 25 แสดงผลลัพธ์เมื่อพบลายนิ้วมือที่บันทึกไว้ก่อนหน้านี้



รูปที่ 26 แสดงผลลัพธ์เมื่อไม่พบลายนิ้วมือที่ค้นหา