ระบบบริหารจัดการสินค้าคงคลังด้วยอาร์เอฟไอดีและเทคโนโลยีคลาวด์

นายอภิชาติ กันสีนวล

วิทยาศาสตรมหาบัณฑิต

สาขาวิทยาการคอมพิวเตอร์

มหาวิทยาลัยเชียงใหม่

พฤษภาคม 2569

ระบบบริหารจัดการสินค้าคงคลังด้วยอาร์เอฟไอดีและเทคโนโลยีคลาวด์

นายอภิชาติ กันสีนวล

วิทยาศาสตรมหาบัณฑิต

สาขาวิทยาการคอมพิวเตอร์

มหาวิทยาลัยเชียงใหม่

พฤษภาคม 2569

ระบบบริหารจัดการสินค้าคงคลังด้วยอาร์เอฟไอดีและเทคโนโลยีคลาวด์

นายอภิชาติ กันสีนวล

การค้นคว้าอิสระนี้เสนอต่อมหาวิทยาลัยเชียงใหม่เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิทยาการคอมพิวเตอร์

มหาวิทยาลัยเชียงใหม่
พฤษภาคม 2569

ระบบบริหารจัดการสินค้าคงคลังด้วยอาร์เอฟไอดีและเทคโนโลยีคลาวด์

นายอภิชาติ กันสีนวล

การค้นคว้าอิสระนี้ได้รับการพิจารณาอนุมัติให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์

คณะกรรมการสอบ

.....ประธานกรรมการ
(รองศาสตราจารย์ ดร.เอกรัฐ บุญเชียง)

.....กรรมการ
(.....)

.....กรรมการ
(.....)

อาจารย์ที่ปรึกษา

.....
(ผู้ช่วยศาสตราจารย์ ดร.ศุภกิจ อาวิพันธุ์)

15 พฤษภาคม 2569

© ลิขสิทธิ์ของมหาวิทยาลัยเชียงใหม่

กิตติกรรมประกาศ

การค้นคว้าแบบอิสระฉบับนี้ สำเร็จลุล่วงไปได้ด้วยดีด้วยความกรุณาและความช่วยเหลือเป็นอย่างยิ่งจาก ผศ.ดร.ศุภกิจ อาวิพันธุ์ อาจารย์ที่ปรึกษา ที่ได้เสียสละเวลาอันมีค่ามาช่วยให้คำปรึกษา แนะนำ ชี้แนะแนวทางที่ถูกต้อง ตลอดจนตรวจแก้ไขข้อบกพร่องต่าง ๆ ด้วยความเอาใจใส่และเกื้อกูลเสมอมา จนทำให้การค้นคว้าแบบอิสระชิ้นนี้มีความสมบูรณ์ ข้าพเจ้าขอขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ข้าพเจ้าขอขอบพระคุณสำนักงานศาลยุติธรรมเป็นอย่างสูง ที่ให้ให้โอกาสและสนับสนุนทุนการศึกษา ซึ่งเป็นส่วนสำคัญในการทำให้การค้นคว้าแบบอิสระฉบับนี้ลุล่วงได้ตามวัตถุประสงค์

ข้าพเจ้าขอกราบขอบพระคุณคณะกรรมการสอบทุกท่าน ที่ได้สละเวลาอันมีค่าในการตรวจสอบ ให้ข้อคิดเห็น และข้อเสนอแนะที่เป็นประโยชน์ต่อการปรับปรุงการค้นคว้าแบบอิสระฉบับนี้ให้มีความถูกต้องและมีคุณค่าทางวิชาการมากยิ่งขึ้น

สุดท้ายนี้ ขอกราบขอบพระคุณบิดา มารดา รวมถึงครอบครัว และเพื่อนร่วมงานทุกท่าน ที่คอยเป็นกำลังใจ ให้การสนับสนุน และความช่วยเหลือในทุก ๆ ด้านตลอดระยะเวลาในการศึกษาและการทำการค้นคว้าแบบอิสระฉบับนี้ คุณค่าและประโยชน์อันใดที่เกิดจากการค้นคว้าแบบอิสระฉบับนี้ ผู้ศึกษาขอมอบเป็นเครื่องบูชาพระคุณแด่บุพการี ครูบาอาจารย์ และผู้มีพระคุณทุกท่านที่มีส่วนร่วมในความสำเร็จครั้งนี้

อภิชาติ กันสีนวล

หัวข้อการค้นคว้าอิสระ	ระบบบริหารจัดการสินค้าคงคลังด้วยอาร์เอฟไอดีและเทคโนโลยีคลาวด์
ผู้เขียน	นาย อภิชาติ กันสีนวล
ปริญญา	วิทยาศาสตรมหาบัณฑิต(วิทยาการคอมพิวเตอร์)
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.ศุภกิจ อาวิพันธุ์

บทคัดย่อ

การบริหารจัดการสินค้าคงคลังในปัจจุบันที่ใช้ระบบบาร์โค้ด หรือ คิวอาร์โค้ด ประสบปัญหาความล่าช้าในการทำงาน เนื่องจากต้องสแกนในระยะใกล้และต้องเห็นรหัสชัดเจน อีกทั้งรหัสที่ติดยังมีโอกาสชำรุดเสียหายได้ง่ายเมื่อเวลาผ่านไป การค้นคว้าแบบอิสระนี้นำเสนอระบบบริหารจัดการสินค้าคงคลังโดยใช้เทคโนโลยีระบุตัวรหัสสินค้าคงคลังด้วยคลื่นวิทยุ ร่วมกับเทคโนโลยีคลาวด์ เพื่อแก้ปัญหาดังกล่าว โดยใช้ อาร์เอฟไอดีแท็ก แบบ UHF และชุดอ่านที่ทำขึ้นโดยใช้ Raspberry Pi ทำให้สามารถอ่านรหัสสินค้าคงคลังได้พร้อมกันหลายชิ้นโดยไม่ต้องเห็นตัวอาร์เอฟไอดีแท็ก ข้อมูลจะถูกประมวลผลและจัดเก็บในระบบคลาวด์ ผ่านแอปพลิเคชัน ที่พัฒนาด้วยเฟรมเวิร์ก Django จากผลการทดลองแสดงให้เห็นว่าระบบอาร์เอฟไอดี มีความรวดเร็วและคุ้มค่าในการบริหารจัดการสินค้าคงคลังที่มีปริมาณมาก เมื่อเทียบกับระบบบาร์โค้ด หรือ คิวอาร์โค้ด แบบดั้งเดิม

คำสำคัญ: คลาวด์เทคโนโลยี, สินค้าคงคลัง, อาร์เอฟไอดี

Independent Study Title	Inventory Management System Using RFID and Cloud Technology
Author	Mr. Apichart Kanseenuan
Degree	Master of Science (Computer Science)
Advisor	Asst.Prof.Dr.Suphakit Awiphan

ABSTRACT

Current inventory management systems relying on Barcode or QR code technologies face operational delays due to the requirement for close-range scanning and a clear line of sight. Furthermore, physical labels are prone to damage over time. This independent study presents an inventory management system integrating Radio Frequency Identification (RFID) and Cloud technology to address these limitations. By utilizing UHF RFID tags and a custom-built reader powered by a Raspberry Pi, the system enables simultaneous identification of multiple items without requiring direct visibility of the tags. Data is processed and stored in the cloud via an application developed using the Django framework. Experimental results demonstrate that the RFID-based system offers superior speed and cost-effectiveness for managing high-volume inventory compared to traditional Barcode or QR code systems.

Keywords: Cloud Technology, Inventory, RFID

สารบัญ

		หน้า
กิตติกรรมประกาศ		ค
บทคัดย่อภาษาไทย		ง
ABSTRACT		จ
สารบัญตาราง		ซ
สารบัญภาพ		ฌ
รายการอักษรย่อ		ญ
อภิธานศัพท์		ฎ
บทที่ 1	บทนำ	1
	1.1 ที่มาและความสำคัญ	1
	1.2 วัตถุประสงค์ของการค้นคว้าแบบอิสระ	2
	1.3 ขอบเขตของการค้นคว้าแบบอิสระ	2
	1.3.1 ขอบเขตด้านฮาร์ดแวร์	2
	1.3.1 ขอบเขตด้านซอฟต์แวร์และสถาปัตยกรรม	2
	1.3.1 ขอบเขตด้านการทดสอบ	3
	1.5 ประโยชน์ที่คาดว่าจะได้รับจากการศึกษา	3
บทที่ 2	แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง	4
	2.1 อาร์เอฟไอดี (RFID : Radio Frequency Identification)	4
	2.2 คอมพิวเตอร์บอร์ดเดี่ยว (Single-Board Computer)	5
	2.3 สถาปัตยกรรมการประมวลผลแบบกลุ่มก้อน (Cloud Computing Architecture)	5
	2.4 เทคนิคการจับคู่ความสัมพันธ์ตำแหน่งและสินค้าแบบบูรณาการ (Integrated Location-Item Binding Technique)	6
	2.5 งานวิจัยที่เกี่ยวข้อง	7
บทที่ 3	วิธีการดำเนินการ	9
	3.1 ศึกษาวิเคราะห์ปัญหาที่เกี่ยวข้อง	9

3.2 เตรียมข้อมูล 9
3.3 จัดทำอุปกรณ์หลัก 10
3.4 จัดทำโปรแกรมทดสอบประสิทธิภาพ 12
3.5 การประเมินประสิทธิภาพของระบบ 12
3.6 กำหนดสิ่งแวดล้อมในการทำการทดลอง 13

บทที่ 4 ผลการทดลองและวิจารณ์ผลการทดลอง 15

บทที่ 5 สรุปผลการทดลอง 18

เอกสารอ้างอิง 19

ภาคผนวก 20
ภาคผนวก ก 21
ภาคผนวก ข 22
ภาคผนวก ค 23

ประวัติผู้เขียน 24

สารบัญตาราง

หน้า

ตารางที่ 4.1 ผลการทดสอบความเร็วระหว่าง อาร์เอฟไอดีแท็ก และ บาร์โค้ด 5 ชิ้น 15

ตารางที่ 4.2 ผลการทดสอบความเร็วระหว่าง อาร์เอฟไอดีแท็ก และ บาร์โค้ด 10 ชิ้น 15

ตารางที่ 4.3 ผลการทดสอบความเร็วระหว่าง อาร์เอฟไอดีแท็ก และ บาร์โค้ด 30 ชิ้น 15

ตารางที่ 4.4 ผลการทดสอบความเร็วระหว่าง อาร์เอฟไอดีแท็ก และ บาร์โค้ด 50 ชิ้น 15

ตารางที่ 4.5 ผลการทดสอบความเร็วระหว่าง อาร์เอฟไอดีแท็ก และ บาร์โค้ด 100 ชิ้น 16

ตารางที่ 4.6 ความเร็วในการอ่านรหัสสินค้าคงคลังเมื่ออยู่ในบรรจุภัณฑ์ 2 ประเภทได้แก่ กล่องกระดาษ และกล่องพลาสติก มีหน่วยเป็นวินาที 17

สารบัญภาพ

		หน้า
ภาพที่ 1.1	ตัวอย่างบาร์โค้ดที่ชำรุดจนไม่สามารถตรวจสอบได้ว่าเป็นรหัสอะไร	1
ภาพที่ 2.1	แสดงอาร์เอฟไอดีแท็กแบบต่าง ๆ เลือกได้ตามรูปแบบการใช้งาน	4
ภาพที่ 2.2	อาร์เอฟไอดีรีดเดอร์โมดูล FM-505	4
ภาพที่ 2.3	แสดงการทำงานของเทคโนโลยีอาร์เอฟไอดี	5
ภาพที่ 2.4	คอมพิวเตอร์บอร์ดเดี่ยว (Raspberry Pi 5)	5
ภาพที่ 3.1	แสดงฐานข้อมูลที่ใช้ในการจัดเก็บสินค้าคงคลัง	10
ภาพที่ 3.2	ส่วนประกอบของชุดอ่านรหัส อาร์เอฟไอดี	11
ภาพที่ 3.3	ต้นแบบชุดอ่านรหัส อาร์เอฟไอดี ด้วย Raspberry Pi 5	11
ภาพที่ 3.4	ตัวอย่างผลลัพธ์จากการทดสอบอ่านรหัสที่ 100 รหัส	12
ภาพที่ 3.5	แสดงการจัดเรียงรหัสที่ใช้ในการทดสอบประสิทธิภาพการอ่าน	13
ภาพที่ 3.6	แสดงการบรรจุสินค้าลงกล่องกระดาษ	14
ภาพที่ 3.7	แสดงการบรรจุสินค้าคงคลังในกล่องพลาสติก	14
ภาพที่ 4.1	แสดงการเปรียบเทียบเวลาที่ใช้ในการตรวจนับสินค้า คงคลัง ระหว่าง RFID Tag และ บาร์โค้ด	17

รายการอักษรย่อ

API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
AWS	Amazon Web Services
DWILL	Django Web Launch Library
IoT	Internet of Things
IR	Infrared
JSON	JavaScript Object Notation
ORM	Object-Relational Mapping
QR Code	Quick Response Code
RFID	Radio Frequency Identification
RTLS	Real-Time Location System
UART	Universal Asynchronous Receiver/Transmitter
UHF	Ultra High Frequency
USB	Universal Serial Bus

อภิธานศัพท์

Barcode	ระบบระบุรหัสสินค้าคงคลังแบบดั้งเดิมที่ต้องใช้หลักการของแสงในการอ่าน เครื่องสแกนจำเป็นต้องมองเห็นตัวรหัสอย่างชัดเจนและต้องตรวจสอบทีละรายการ
Bulk Reading	การอ่านแบบกลุ่ม คือความสามารถเด่นของเทคโนโลยีอาร์เอฟไอดีที่สามารถอ่านและตรวจสอบรหัสสินค้าคงคลังได้พร้อมกันหลายรหัสในคราวเดียว ทำให้ประหยัดเวลา
Cloud Technology	เทคโนโลยีคลาวด์ เป็นระบบสำหรับจัดเก็บฐานข้อมูลและประมวลผลผ่านอินเทอร์เน็ต ช่วยลดความยุ่งยากในการดูแลฮาร์ดแวร์เครื่องแม่ข่าย และเข้าถึงได้จากหลากหลายอุปกรณ์
Code 128	รูปแบบมาตรฐานของรหัสบาร์โค้ดที่สามารถระบุอักขระ ASCII ได้ครบทั้ง 128 ตัวอักษร นำมาใช้เพื่อรองรับการแปลงรหัสที่อ่านได้จากอาร์เอฟไอดีแท็ก
Continuous Sampling	การอ่านซ้ำแบบต่อเนื่อง เป็นเทคนิคที่ประยุกต์ใช้ความเร็วของโมดูลอาร์เอฟไอดี เพื่อให้ระบบทำการตรวจสอบความถูกต้องของรหัสซ้ำๆ ในเสี้ยววินาที ช่วยลดอัตราการอ่านผิดพลาด
Django Framework	เฟรมเวิร์กสำหรับการพัฒนาเว็บแอปพลิเคชันที่เขียนด้วยภาษาไพทอน นำมาใช้ในการสร้าง API และจัดการฐานข้อมูลบนคลาวด์
JSON	JavaScript Object Notation รูปแบบมาตรฐานของข้อมูลที่ระบบใช้แพ็กข้อมูลการเข้า-ออกของสินค้า เพื่อส่งผ่านช่องทาง API ขึ้นไปจัดเก็บบนคลาวด์
Kivy Framework	เฟรมเวิร์กที่รองรับการทำงานข้ามแพลตฟอร์ม นำมาใช้พัฒนาแอปพลิเคชันส่วนติดต่อผู้ใช้งาน ให้รองรับทั้งคอมพิวเตอร์, Android และ iOS
QR Code	ระบบระบุรหัสสินค้าคงคลังแบบ 2 มิติที่ทำงานด้วยหลักการแสง มีข้อจำกัดด้านความล่าช้าเพราะต้องสแกนทีละรายการ และมีโอกาสชำรุดเสียหายได้ง่าย
Raspberry Pi 5	บอร์ดไมโครคอมพิวเตอร์ขนาดเล็กที่ทำหน้าที่เป็นหน่วยประมวลผลหลัก

	สำหรับชุดเครื่องอ่านอาร์เอฟไอดีต้นแบบ
RESTful API	ช่องทางการสื่อสารของระบบซอฟต์แวร์ที่สร้างขึ้นเพื่อให้อุปกรณ์สามารถส่งข้อมูลสินค้าขึ้นไปอัปเดตบนคลาวด์ได้แบบเรียลไทม์
RFID	เทคโนโลยีการระบุข้อมูลด้วยคลื่นวิทยุ (อาร์เอฟไอดี) มีความทนทานสูง ซ่อนในบรรจุภัณฑ์ได้ และอ่านรหัสได้ทีละหลายรายการ
RFID Tag	ป้ายระบุรหัสสินค้าคงคลังที่รับและส่งสัญญาณคลื่นวิทยุตอบกลับมายังเครื่องอ่าน (ในงานนี้ใช้แบบ Passive Sticker ไม่มีแบตเตอรี่ในตัว)
UART	โปรโตคอลการสื่อสารข้อมูลแบบอนุกรม ใช้เป็นช่องทางเชื่อมต่อและสื่อสารข้อมูลระหว่างโมดูลอ่านอาร์เอฟไอดีกับบอร์ด Raspberry Pi
UHF	Ultra High Frequency คลื่นวิทยุย่านความถี่สูงพิเศษ ซึ่งเป็นย่านความถี่เฉพาะที่ระบบอาร์เอฟไอดีแท็กในโครงงานนี้เลือกใช้งาน

บทที่ 1

บทนำ

บทนี้จะกล่าวถึงภาพรวมของพื้นฐาน วัตถุประสงค์ ขอบเขต และระเบียบวิธีศึกษา โดยได้นำเสนอการนำ เทคโนโลยีอาร์เอฟไอดี และระบบคลาวด์ มาใช้เพื่อเพิ่มประสิทธิภาพในการบริหารจัดการสินค้าคงคลัง เทียบกับการใช้ บาร์โค้ด และ คิวอาร์โค้ด

1.1 ที่มาและความสำคัญ

ในการบริหารสินค้าคงคลังในปัจจุบันนิยมใช้บาร์โค้ด และ คิวอาร์โค้ด ระบุตัวสินค้าคงคลัง ขั้นตอนการตรวจนับมักจะต้องใช้เวลาล่าช้าเพราะต้องตรวจสอบทีละรายการ และตำแหน่งของรหัสอาจจะไม่ได้อยู่ในจุดเดียวกัน ส่งผลเสียเวลาจัดวางตำแหน่งเพื่อให้ตรวจสอบได้ด้วยเครื่องอ่าน อีกทั้งยังต้องมองเห็นรหัสอย่างชัดเจน จึงจะสามารถระบุรหัสได้ ทำให้ไม่สามารถตรวจสอบสินค้าคงคลังที่อยู่ในกล่องปิดตายได้ อีกปัญหาหนึ่งก็คือป้ายที่ระบุรหัสส่วนมากจะเป็นกระดาษ ที่มักจะชำรุด หลุดลอก หรือเลือนหายได้ง่ายจากการใช้งาน หรือโดนความชื้นดังภาพที่ 1.1 ส่งผลต้องเสียทรัพยากรและเวลาในการจัดหารหัสใหม่มาทดแทน ยิ่งไปกว่านั้น ระบบเดิมยังพึ่งพาการจัดเก็บข้อมูลบนเซิร์ฟเวอร์ท้องถิ่น ซึ่งทำให้เกิดภาระค่าใช้จ่ายสูง ทั้งด้านอุปกรณ์ พลังงานไฟฟ้า และค่าบำรุงรักษาอย่างต่อเนื่อง จากปัญหาและข้อจำกัดที่กล่าวมา การค้นคว้าอิสระฉบับนี้จึงมีวัตถุประสงค์เพื่อพัฒนาต้นแบบระบบบริหารจัดการสินค้าคงคลังที่มีประสิทธิภาพสูงขึ้น เพื่อนำมาประยุกต์ใช้งานแทนระบบเดิม

ภาพที่ 1.1 ตัวอย่างบาร์โค้ดที่ชำรุดจนไม่สามารถตรวจสอบได้ว่าเป็นรหัสอะไร

1.2 วัตถุประสงค์ของการค้นคว้าแบบอิสระ

1) เพื่อพัฒนาระบบบริหารจัดการสินค้าคงคลังด้วยการนำเทคโนโลยีอาร์เอฟไอดีและเทคโนโลยีคลาวด์มาประยุกต์ใช้
2) เพื่อออกแบบและสร้างชุดเครื่องอ่านอาร์เอฟไอดีต้นแบบที่พกพาได้ โดยใช้ Raspberry Pi เป็นหน่วยประมวลผลหลัก
3) เพื่อประเมินและเปรียบเทียบประสิทธิภาพด้านความเร็วในการอ่านข้อมูลระหว่างระบบเดิมที่ใช้บาร์โค้ด และ อาร์เอฟไอดีด้วยชุดเครื่องอ่านที่ได้จัดทำขึ้น

1.3 ขอบเขตของการค้นคว้าแบบอิสระ

1.3.1 ขอบเขตด้านอาร์ดแวร์

1) หน่วยประมวลผลหลักใช้บอร์ดคอมพิวเตอร์แบบเดี่ยว (Single-Board Computer) รุ่น Raspberry Pi 5 ในการควบคุมการทำงานประประมวลผลของชุดอุปกรณ์ต้นแบบ
2) ชุดรับ-ส่งสัญญาณคลื่นวิทยุใช้ RFID Module รุ่น FM-505 โดยกำหนดให้เชื่อมต่อและสื่อสารข้อมูลกับหน่วยประมวลผลหลักผ่านโปรโตคอลอนุกรม UART (Universal Asynchronous Receiver-Transmitter)
3) ป้ายรหัสสินค้าใช้อาร์เอฟไอดีแท็กแบบไม่มีแหล่งจ่ายไฟในตัว (Passive Tag) ย่านความถี่สูงพิเศษ UHF (Ultra-High Frequency) รุ่น Alien 9662 U8 ในรูปแบบของสติกเกอร์สำหรับติดตั้งบนตัวสินค้า
4) ออกแบบชุดอุปกรณ์พกพาใช้แบตเตอรี่ขนาด 12V ร่วมกับอุปกรณ์ปรับแรงดันไฟ (Step-down Buck Converter) ให้คงที่ 5V เพื่อจ่ายไฟให้กับระบบ และมีหน้าจอแสดงผลแบบสัมผัสได้ขนาด 7 นิ้ว

1.3.2 ขอบเขตด้านซอฟต์แวร์และสถาปัตยกรรม

1) พัฒนาโปรแกรมควบคุมการทำงานของฮาร์ดแวร์ด้วยภาษาไพทอน (Python)
2) พัฒนาระบบจัดการฐานข้อมูลบนคลาวด์เฟรมเวิร์ก Django เพื่อทำหน้าที่จัดการข้อมูลสินค้าคงคลังและสื่อสารข้อมูลสองทางผ่านสถาปัตยกรรม RESTful API ในรูปแบบโครงสร้างข้อมูล JSON
3) ส่วนติดต่อของข้อมูลใช้พัฒนาด้วยเฟรมเวิร์ก Kivy (Cross-platform) เพื่อให้รองรับการใช้งานบนระบบปฏิบัติการ Android, iOS และคอมพิวเตอร์

1.3.3 ขอบเขตด้านการทดสอบ

1) ทดสอบเปรียบเทียบระยะเวลาในการอ่านระหว่างเครื่องอ่าน USB กับชุดอ่านรหัสอาร์เอฟไอดีที่จัดทำขึ้น

2) ประเมินประสิทธิภาพคด้านความเร็วในการตรวจนับและประมวลผลข้อมูล โดยใช้จำนวนตัวอย่าง 5, 10, 30, 50 และ 100 รหัส (ชิ้น) เพื่อหาค่าเฉลี่ยของเวลาที่ใช้

3) ทดสอบประสิทธิภาพในการอ่านคลื่นวิทยุระหว่างบรรจุภัณฑ์ที่เป็นกล่องกระดาษ และ กล่องพลาสติก

1.4 ประโยชน์ที่คาดว่าจะได้รับจากการศึกษา

1.5.1 สามารถสร้างต้นแบบระบบบริหารการจัดการสินค้าคงคลังด้วยอาร์เอฟไอดีร่วมกับเทคโนโลยีคลาวด์ ซึ่งสามารถนำไปเป็นแนวทางการพัฒนาระบบบริหารจัดการสินค้าคงคลังเดิมได้

1.5.2 สามารถเปรียบเทียบข้อดีข้อเสียสำหรับการใช้บาร์โค้ด หรือ คิวอาร์โค้ด กับ อาร์เอฟไอดี เพื่อใช้เป็นข้อมูลเชิงประจักษ์ประกอบการตัดสินใจเชิงนโยบายในการปรับเปลี่ยนเทคโนโลยี

1.5.3 สามารถนำไปพัฒนากับระบบควบคุมครุภัณฑ์หรือเอกสารของสำนักงานศาลยุติธรรมที่ส่วนใหญ่ยังใช้ระบบบาร์โค้ดอยู่ เพื่อยกระดับความรวดเร็วและลดข้อผิดพลาดในกระบวนการตรวจนับปัจจุบัน

บทที่ 2

แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

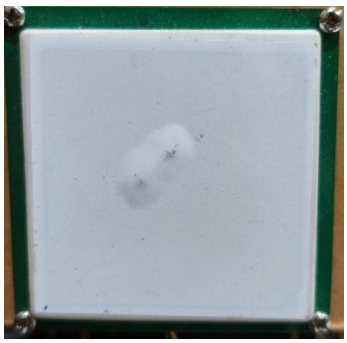
ในการค้นคว้าแบบอิสระนี้ให้ความสำคัญในการปรับปรุงเพิ่มประสิทธิภาพในการจัดการสินค้าคงคลังที่ใช้ในปัจจุบันและยังเพิ่มทางเลือกในการใช้งานเทคโนโลยีสมัยใหม่เพื่อมาปรับปรุงระบบงานเดิม

2.1 อาร์เอฟไอดี (RFID : Radio Frequency Identification) เป็นการระบุตัวตนด้วยคลื่นความถี่วิทยุ โดยจะใช้คลื่นวิทยุในการส่งและรับข้อมูลจากระยะไกล โดยไม่จำเป็นต้องสัมผัส หรืออยู่ในแนวสายตา ประกอบด้วย 2 ส่วนดังนี้

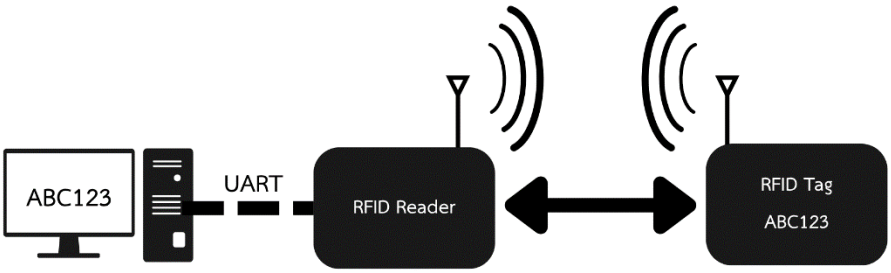
1) อาร์เอฟไอดีแท็ก (RFID Tags) ประกอบด้วยชิปประมวลผลและเสาอากาศ ทำหน้าที่เก็บรหัสประจำตัว

ภาพที่ 2.1 แสดงอาร์เอฟไอดีแท็กแบบต่าง ๆ เลือกใช้ตามรูปแบบการใช้งาน

2) อาร์เอฟไอดีรีดเดอร์ (RFID Reader) ทำหน้าที่ส่งคลื่นวิทยุเพื่อกระตุ้นอาร์เอฟไอดีแท็ก เพื่อให้แท็กส่งสัญญาณข้อมูลกลับมา ดังภาพที่ 2.3



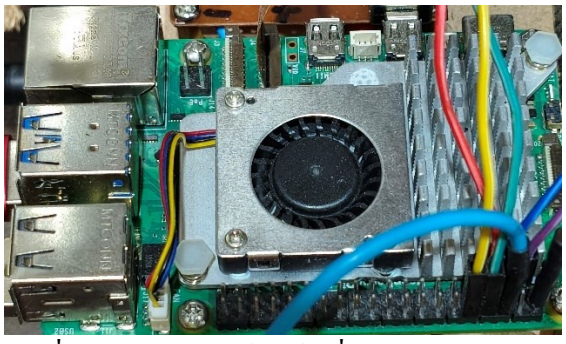
ภาพที่ 2.2 อาร์เอฟไอดีรีดเดอร์โมดูล FM-505



ภาพที่ 2.3 แสดงการทำงานของเทคโนโลยีอาร์เอฟไอดี

2.2 คอมพิวเตอร์บอร์ดเดี่ยว (Single-Board Computer) เป็นคอมพิวเตอร์ขนาดเล็กสำหรับการประมวลผลใช้พลังงานไฟฟ้าในการทำงานน้อยติดต่อสื่อสารกับอุปกรณ์ต่อพ่วงได้หลากหลาย นิยมนำมาประยุกต์กับระบบงานเดิมเพื่อเพิ่มความคล่องตัว

ในการค้นคว้าอิสระนี้จะใช้ Raspberry Pi 5 ซึ่งเป็นคอมพิวเตอร์บอร์ดเดี่ยวที่นิยมใช้กันอย่างแพร่หลายมาทำเป็นชุดควบคุมอาร์เอฟไอดีและทำการประมวลผลแล้วส่งต่อข้อมูลผ่าน RESTful API ไปยังคลาวด์เพื่อจัดเก็บข้อมูล



ภาพที่ 2.4 คอมพิวเตอร์บอร์ดเดี่ยว (Raspberry Pi 5)

2.3 **สถาปัตยกรรมการประมวลผลแบบกลุ่มก้อน (**Cloud Computing Architecture) เป็นรูปแบบการให้บริการทรัพยากรคอมพิวเตอร์ผ่านอินเทอร์เน็ต โดยผู้ใช้สามารถจัดการกับทรัพยากรเช่น เซิร์ฟเวอร์ ฐานข้อมูล ซอฟต์แวร์ ได้ตามต้องการ โดยที่ไม่ต้องดูแลโครงสร้างพื้นฐานเอง ทำให้ลดต้นทุนด้านฮาร์ดแวร์ บุคลากรและค่าไฟฟ้า รองรับการขยายระบบได้รวดเร็ว แบ่งรูปแบบออกเป็น 3 แบบดังนี้

1) Infrastructure as a Service (IaaS) ผู้ให้บริการจะจัดเตรียมทรัพยากรทั้งหมดไว้ให้ เช่น เครื่องเสมือน หน่วยประมวลผล หน่วยความจำ พื้นที่จัดเก็บข้อมูล และเครือข่าย ผู้ใช้บริการมีหน้าที่ติดตั้งระบบปฏิบัติการ และแอปพลิเคชันเอง

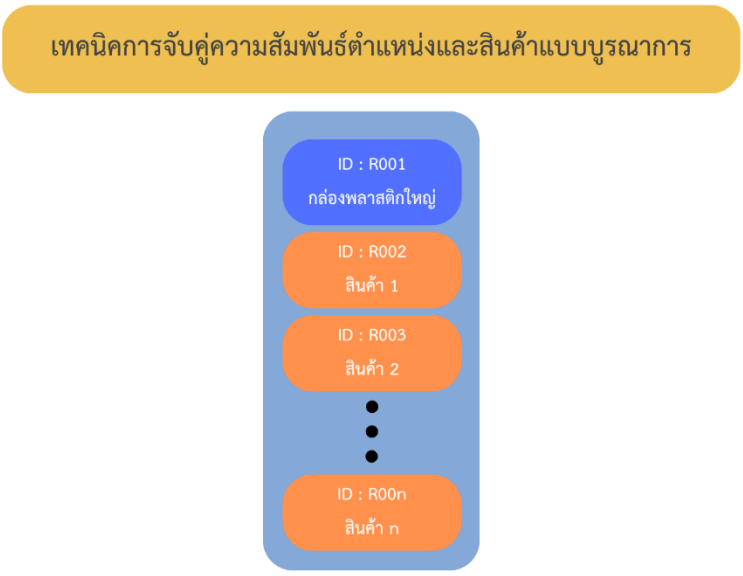
2) Platform as a Service (PaaS) ผู้ให้บริการจะจัดเตรียมโครงสร้างพื้นฐานและเครื่องมือไว้ให้พร้อมใช้งานผู้ใช้สามารถสร้าง ทดสอบ และปรับใช้แอปพลิเคชันได้โดยไม่ต้องดูแลโครงสร้างพื้นฐานเลย

3) Software as a Service (SaaS) เป็นซอฟต์แวร์สำเร็จรูปที่ผู้ให้บริการทำหน้าที่ดูแลทั้งหมดไม่ว่าจะเป็นด้านแอปพลิเคชันและโครงสร้างพื้นฐาน ผู้ใช้งานสามารถใช้งานได้ทันทีไม่ต้องติดตั้งและบำรุงรักษาเอง

ภาพที่ 2.5 แสดงสถาปัตยกรรมแบบต่าง ๆที่มีในปัจจุบัน

2.4 เทคนิคการจับคู่ความสัมพันธ์ตำแหน่งและสินค้าแบบบูรณาการ (Integrated Location-Item Binding Technique) ในการบริหารสินค้าคงคลังข้อจำกัดในการปฏิบัติงานล่าช้าไม่ได้เกิดจากความเร็วในการอ่านรหัสสินค้าอย่างเดียว แต่เกิดจากเวลาที่สูญเสียไปในการให้ผู้ใช้เลือกตำแหน่งที่เก็บสินค้าก่อนแล้วถึงจะทำการอ่านรหัสสินค้า งานวิจัยนี้ได้นำเสนอเทคนิคในการบริหารสินค้าคงคลังแบบการจับคู่ความสัมพันธ์ตำแหน่งและสินค้าแบบบูรณาการ (Integrated Location-Item Binding Technique) โดยจะประยุกต์การใช้อาร์เอฟไอดีแท็กทำหน้าที่ 2 บทบาทคือ ระบุตำแหน่งสินค้า และระบุตัวสินค้าเอง ด้วยความสามารถของการอ่านแบบกลุ่ม (Bulk Reading) ของเทคโนโลยีอาร์เอฟไอดี เมื่อผู้ใช้ทำการตรวจสินค้าบริเวณชั้นวาง หรือบรรจุภัณฑ์อยู่ คลื่นวิทยุจะครอบคลุมและอ่านข้อมูล

ทั้งรหัสตำแหน่งที่เก็บและรหัสสินค้าทั้งหมดในคราวเดียวกันและทำการส่งข้อมูลที่ได้ผ่าน RESTful API ขึ้นสู่คลาวด์ เทคนิคนี้จะช่วยลดระยะเวลาการทำงาน และยังได้ความแม่นยำขึ้นอีกด้วย



ภาพที่ 2.6 แสดงตัวอย่างการจัดสินค้าและการกำหนดรหัส

2.5 งานวิจัยที่เกี่ยวข้อง

การค้นคว้าแบบอิสระนี้เป็นการพัฒนาต้นแบบของระบบบริหารสินค้าคงคลังจากเดิมที่ใช้ระบบบาร์โค้ดเป็นอารเอฟไอดีรวมทั้งการเปลี่ยนโครงสร้างการจัดเก็บข้อมูลโดยใช้เทคโนโลยีคลาวด์เป็นหลัก งานวิจัยที่เกี่ยวข้องจึงเป็นตัวอย่างการนำทั้งสองเทคโนโลยีนี้มาประยุกต์เพื่อเพิ่มประสิทธิภาพจากระบบเดิมที่ทำอยู่

Intelligent RFID Based Library Management System เป็นงานที่พัฒนาระบบจัดการห้องสมุดโดยใช้อาร์เอฟไอดีร่วมกับ Raspberry Pi เพื่อจัดการงานของห้องสมุด เช่น ยืม คืน ค้นหาหนังสือ คำนวณค่าปรับ สิ่งที่ได้จากระบบคือการลดข้อผิดพลาดจากการทำงานของมนุษย์ ป้องกันการสูญหายของหนังสือและเพิ่มความสามารถในการวิเคราะห์หนังสือที่มีการใช้มากหรือน้อยเพื่อตัดสินใจจัดเก็บหรือทำลายทิ้ง [1]

Automated Billing System using RFID and Cloud เป็นระบบคิดเงินสำหรับรถขนโดยใช้เครื่องอ่านอาร์เอฟไอดีและเซนเซอร์อินฟราเรด(IR) ติดตั้งที่รถเข็นเพื่อตรวจสอบสินค้าที่ใส่ลงไป ควบคุมระบบด้วย Raspberry Pi และส่งข้อมูลไปจัดเก็บไว้ที่ AWS ซึ่งเป็นบริการคลาวด์ของ Amazon สิ่งที่ได้จากระบบนี้คือช่วยลดเวลาการต่อคิวเพื่อชำระสินค้า ลดข้อผิดพลาดในการคิดเงินจากพนักงาน และลูกค้าที่ใช้งานจะได้รับประสบการณ์ที่ดีในการใช้บริการไปพร้อมกัน [2]

Localization of Health Center Assets Through an IoT Environment (LoCATE) เป็นระบบที่พัฒนามาเพื่อติดตามตำแหน่งสินทรัพย์ของศูนย์สุขภาพแบบเรียลไทม์ โดยอาศัยโครงสร้างสัญญาณ Wi-Fi (802.11) ที่มีอยู่แล้วมาช่วยเก็บข้อมูลการเคลื่อนไหว โดยใช้ Raspberry Pi เป็นอุปกรณ์หลักในการประมวลผล และจัดเก็บข้อมูลไว้ที่ AWS ซึ่งเป็นบริการคลาวด์ของ Amazon ทำให้ผู้ใช้งานตรวจสอบข้อมูลผ่านหน้าแอปพลิเคชันได้แบบเรียลไทม์ [3]

Smart Inventory Management System for Hardware Warehouse Using IoT เป็นการพัฒนาระบบคลังสินค้าโดยใช้บอร์ด ESP32 ทำงานร่วมกับอาร์เอฟไอดี เซนเซอร์อัลตราโซนิกวัดปริมาณวัสดุที่เหลือ บันทึกข้อมูลไว้ที่ Google Sheets และแสดงข้อมูลและแจ้งเตือนปริมาณวัสดุคงเหลือต่ำกว่าที่ระบุแบบเรียลไทม์ผ่านผู้ให้บริการคลาวด์ Blynk [4]

บทที่ 3

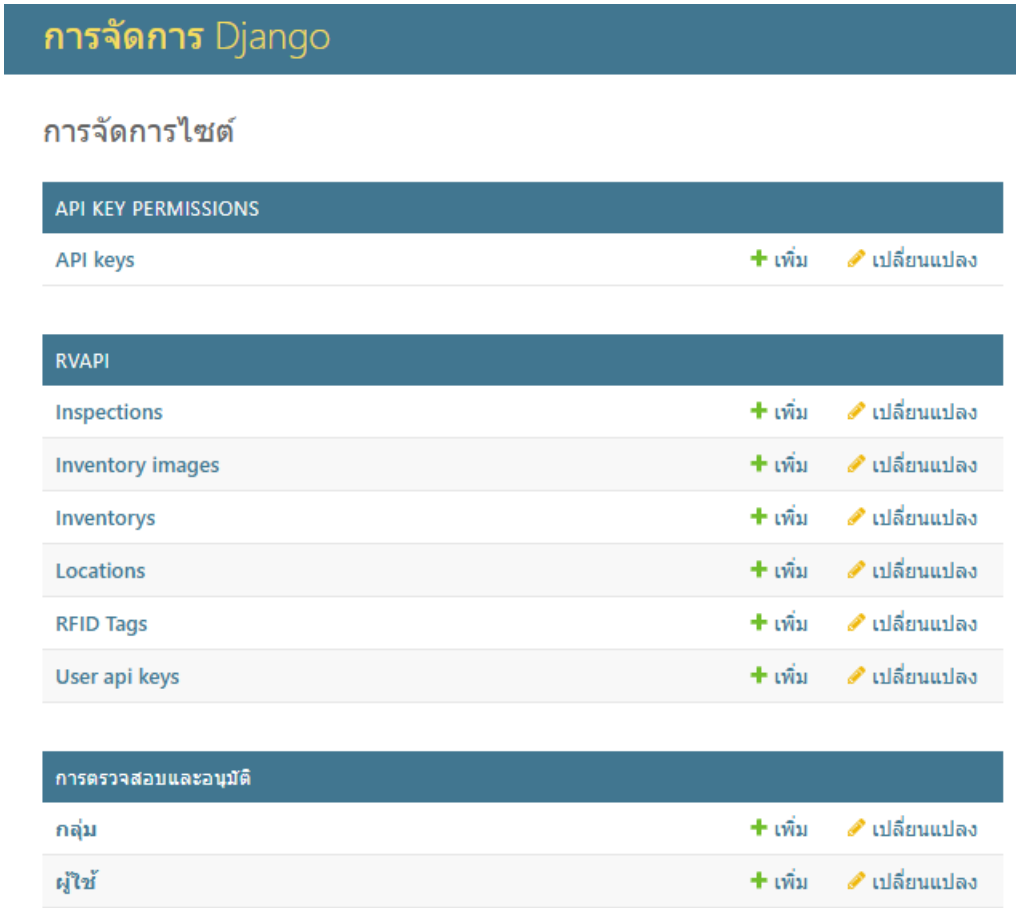
วิธีการดำเนินการ

3.1 ศึกษาวิเคราะห์ปัญหาที่เกี่ยวข้อง

จากปัญหาการใช้งาน บาร์โคด และ คิวอาร์โคด ในการควบคุมสินค้าคงคลังมีปัญหาในเรื่องตัวรหัสเสียหาย ชำรุด และเลือนตามกาลเวลา ทำให้การตรวจสอบแต่ละครั้งใช้เวลานาน ทำให้เสียทรัพยากรที่มีไปโดยไม่จำเป็น เช่นตัวรหัสที่ต้องมีการจัดเตรียมใหม่เมื่อมีการชำรุดหรือเลือนจนอ่านค่าไม่ได้ เป็นต้น

3.2 เตรียมข้อมูล

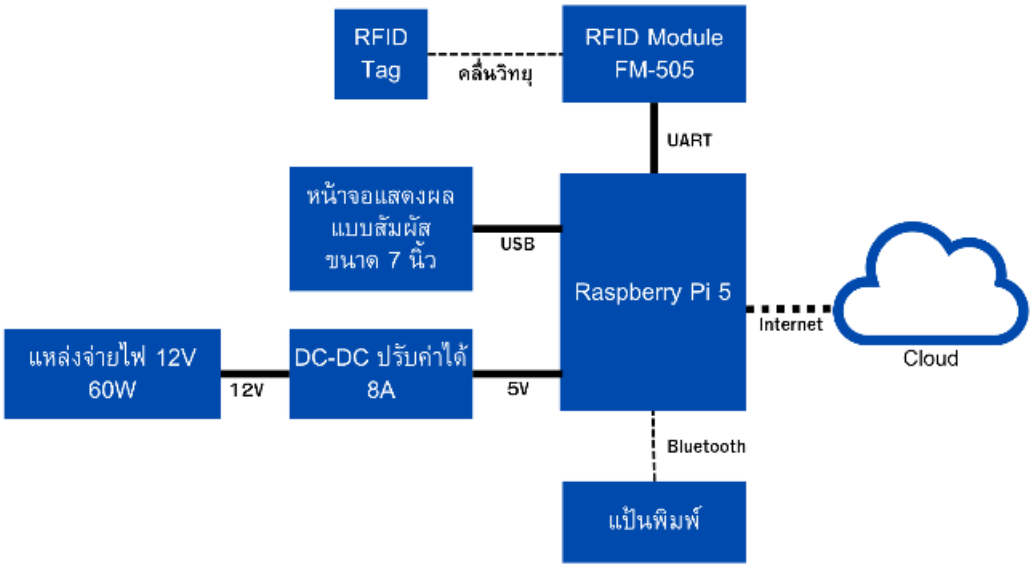
ทำการอ่านรหัสจาก อาร์เอฟไอดีแท็ก เพื่อนำมาทำ บาร์โค้ด โดยจะใช้รหัส บาร์โค้ด 128 (Code 128) ซึ่งเป็นรหัสที่ระบุรหัส ASCII ได้ครบทั้ง 128 ตัว ซึ่งเพียงพอกับการนำรหัสจาก อาร์เอฟไอดีแท็ก มาใช้งาน จากนั้นทำการออกแบบฐานข้อมูลเพื่อให้เก็บข้อมูลที่จำเป็นในการบริหารสินค้าคงคลังเช่น รหัสสินค้าคงคลัง, ชื่อ, สถานที่จัดเก็บ ดังแสดงได้ตามภาพที่ 3.1 และทำการสร้าง API ในการเชื่อมต่อฐานข้อมูลด้วย Django โดยในการศึกษาครั้งนี้จะทำการเก็บระบบไว้กับผู้ให้บริการคลาวด์ซึ่งสามารถลดความยุ่งยากในการจัดการฮาร์ดแวร์ เช่น เครื่องแม่ข่ายคอมพิวเตอร์ อุปกรณ์เครือข่าย ฯลฯ ประหยัดค่าใช้จ่าย เช่นค่าไฟฟ้า ค่าดูแลอุปกรณ์ รวมถึงค่าใช้จ่ายของบุคลากรที่ต้องคอยดูแลอุปกรณ์ให้สามารถใช้งานได้ตลอดเวลา สามารถเพิ่มหรือลดขนาดของบริการได้ตลอดเวลา อีกทั้งยังมีความปลอดภัยในระดับเป็นมาตรฐาน [5]



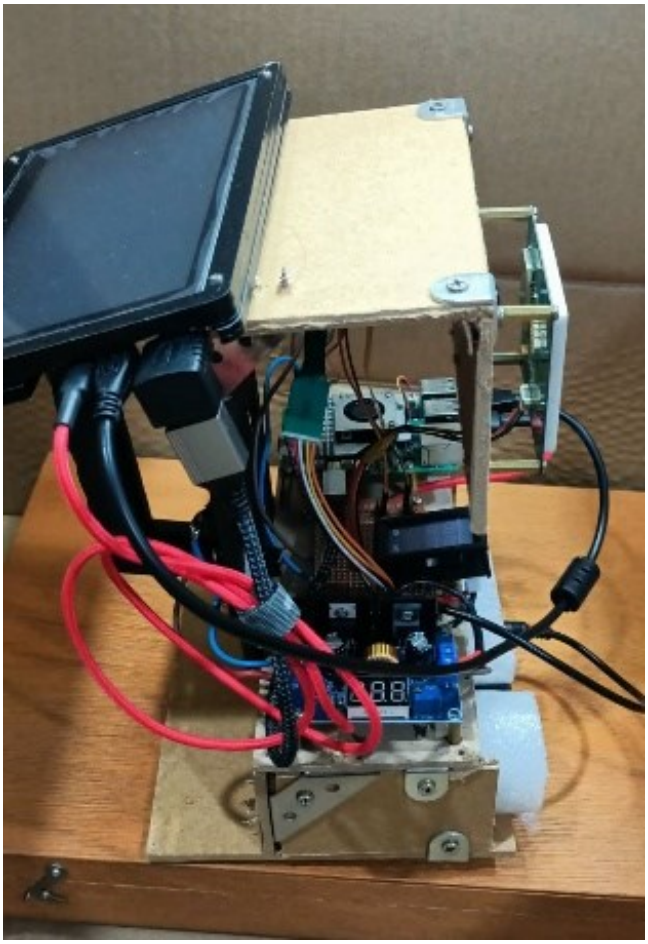
ภาพที่ 3.1 แสดงฐานข้อมูลที่ใช้ในการจัดเก็บสินค้าคงคลัง

3.3 ออกแบบอุปกรณ์หลัก

ในการใช้งาน RFID Module จะต้องมีอุปกรณ์ที่อ่านค่าที่ส่งออกมาจากอุปกรณ์ ในการศึกษาครั้งนี้จะใช้ Raspberry Pi 5 เป็นอุปกรณ์หลักในการศึกษา โดยการสื่อสารจะทำการผ่านโปรโตคอล UART เป็นการสื่อสารแบบอนุกรม มีรหัสควบคุมการอ่านเฉพาะในการสั่งงานและควบคุม พัฒนาระบบด้วยภาษาไพทอน ตัวอุปกรณ์ต้นแบบใช้ Raspberry Pi 5 ในการควบคุมการอ่าน อาร์เอฟไอดีแท็ก และการติดต่อสื่อสารกับระบบคลาวด์ตาม API ที่จัดทำขึ้น ตัวอุปกรณ์มีแหล่งพลังงานจากแบตเตอรี่ 12V ที่ทำการแปลงแรงดันด้วยอุปกรณ์ปรับแรงดันให้ได้ 5V ตามที่ Raspberry Pi 5 ต้องการ และติดตั้งหน้าจอแบบสัมผัสได้ ขนาด 7 นิ้ว ไว้เพื่อแสดงผลลัพธ์การทำงานของระบบงาน ดังแสดงได้ตามภาพที่ 3.2 และชุดต้นแบบแสดงได้ในภาพที่ 3.3



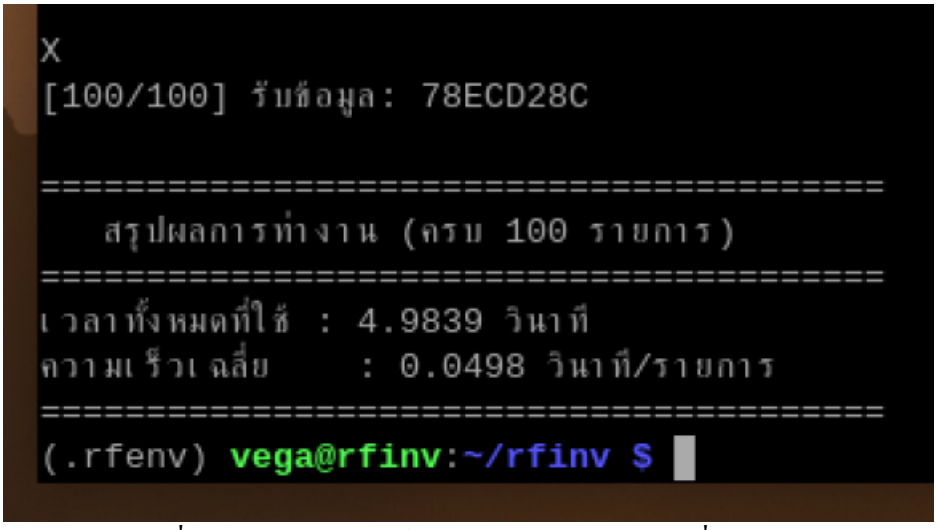
ภาพที่ 3.2 ส่วนประกอบของชุดอ่านรหัส อาร์เอฟไอดี



ภาพที่ 3.3 ต้นแบบชุดอ่านรหัส อาร์เอฟไอดี ด้วย Raspberry Pi 5

3.4 จัดทำโปรแกรมทดสอบประสิทธิภาพ

จัดทำโปรแกรมใช้ทดสอบประสิทธิภาพการอ่านข้อมูลด้วย อาร์เอฟไอดี เทียบกับ บาร์โค้ด โดยจะใช้อุปกรณ์อ่านรหัส บาร์โค้ด แบบ USB เทียบกับอุปกรณ์ที่พัฒนาขึ้น โดยตัวโปรแกรมจะรับค่าพารามิเตอร์ 1 ตัว คือจำนวนข้อมูลที่ต้องการทดสอบ จากนั้นทำการอ่านรหัสสินค้าคงคลังให้ได้ครบตามจำนวนที่ระบุรายละเอียดของการตรวจสอบรหัสบาร์โค้ดอยู่ที่ภาคผนวก ก และ ภาคผนวก ข เมื่อโปรแกรมทำงานเรียบร้อยจะแสดงผลเป็นระยะเวลาที่ใช้ และเวลาเฉลี่ยต่อการเรียกข้อมูลต่อรหัสให้ตามภาพที่ 3.4



ภาพที่ 3.4 ตัวอย่างผลลัพธ์จากการทดสอบอ่านรหัสที่ 100 รหัส

3.5 การประเมินประสิทธิภาพของระบบ

การประเมินผลจะใช้เวลาเฉลี่ยในการอ่านรหัสเทียบกัน โดยออกแบบการอ่านรหัสในจำนวนที่แตกต่างกัน ในการศึกษานี้จะใช้จำนวน 5, 10, 30, 50 และ 100 รหัส และทดสอบประสิทธิภาพในการอ่านในบรรจุภัณฑ์ที่ปิดตายเช่นกล่องกระดาษทึบแสงดังแสดงได้ในภาพที่ 3.6 และกล่องพลาสติกใสแบบมีฝาปิดแสดงได้ในภาพที่ 3.7

คำนวณได้จากสูตรการหาค่าเฉลี่ยในการอ่านรหัสสินค้าคงคลัง (Average Time)

$$\bar{t} = \frac{\sum_{i=1}^{n} t_i}{n}$$

โดยกำหนดให้:

$\bar{t}$ คือ เวลาเฉลี่ยในการอ่าน มีหน่วยเป็นวินาที

t_i คือ เวลาที่ใช้ในการทดสอบรอบที่ i

n คือ จำนวนรอบที่ทำการทดสอบ $(n = 5)$

การหาความเร็วในการอ่านข้อมูล (Reading Speed)

$$Speed = \frac{N}{\bar{t}}$$

โดยกำหนดให้:

$Speed$ คือ ความเร็วในการอ่านรหัสสินค้าคงคลังมีหน่วยเป็น (รหัส/วินาที)

N คือ จำนวนรหัสสินค้าคงคลังที่นำมาทดสอบ (5, 10, 30, 50, 100 รหัส)

$\bar{t}$ คือ เวลาเฉลี่ยจากการคำนวณ

3.6 กำหนดสิ่งแวดล้อมในการทำการทดลอง

ในการทดลองจะนำบาร์โค้ด และ อาร์เอฟไอดีแท็ก ที่มีรหัสเดียวกัน ติดไว้กับพื้นราบ จากนั้นจะทำการเรียงรหัสให้มองเห็นชัดเจนทั้งบาร์โค้ด และ อาร์เอฟไอดีแท็ก เพื่อขจัดปัจจัยที่จะมีผลต่อความเร็วในการอ่านให้เหลือน้อยที่สุด เพื่อวัดความเร็วที่ได้จากการอ่านด้วยเครื่องอ่านบาร์โค้ดและ ชุดอ่านอาร์เอฟไอดีที่ได้จัดทำขึ้น โดยตั้งระยะการอ่านไว้ที่ 50 เซนติเมตร และหันตัวอุปกรณ์ขนานกับรหัสที่ติดกับสินค้าคงคลังที่สุดตามภาพที่ 3.5

ภาพที่ 3.5 แสดงการจัดเรียงรหัสที่ใช้ในการทดสอบประสิทธิภาพการอ่าน

ภาพที่ 3.6 แสดงการบรรจุสินค้าลงกล่องกระดาษ

ภาพที่ 3.7 แสดงการบรรจุสินค้าคงคลังในกล่องพลาสติก

บทที่ 4

ผลการทดลองและวิจารณ์ผลการทดลอง

ผลการทดลองประสิทธิภาพในการอ่านรหัสระหว่าง อาร์เอฟไอดีแท็ก และ บาร์โค้ด ทั้ง 5 แบบ แสดงได้ตามนี้

ตารางที่ 4.1 ผลการทดสอบความเร็วระหว่าง อาร์เอฟไอดีแท็ก และ บาร์โค้ด 5 ชิ้น

ประเภท	ครั้งที่					เฉลี่ย
	1	2	3	4	5	
อาร์เอฟไอดี	0.81	0.75	0.58	0.63	0.92	0.74
บาร์โค้ด	9.34	7.24	8.62	8.68	8.34	8.37

ตารางที่ 4.2 ผลการทดสอบความเร็วระหว่าง อาร์เอฟไอดีแท็ก และ บาร์โค้ด 10 ชิ้น

ประเภท	ครั้งที่					เฉลี่ย
	1	2	3	4	5	
อาร์เอฟไอดี	0.86	1.21	0.73	0.92	1.53	1.05
บาร์โค้ด	14.26	13.37	12.68	13.49	14.35	13.63

ตารางที่ 4.3 ผลการทดสอบความเร็วระหว่าง อาร์เอฟไอดีแท็ก และ บาร์โค้ด 30 ชิ้น

ประเภท	ครั้งที่					เฉลี่ย
	1	2	3	4	5	
อาร์เอฟไอดี	5.24	4.69	6.62	5.42	5.65	5.52
บาร์โค้ด	43.28	39.42	42.36	37.49	38.24	40.16

ตารางที่ 4.4 ผลการทดสอบความเร็วระหว่าง อาร์เอฟไอดีแท็ก และ บาร์โค้ด 50 ชิ้น

ประเภท	ครั้งที่					เฉลี่ย
	1	2	3	4	5	
อาร์เอฟไอดี	7.43	8.94	9.71	8.65	8.21	8.59
บาร์โค้ด	72.43	82.38	76.46	74.85	78.69	76.96

ตารางที่ 4.5 ผลการทดสอบความเร็วระหว่าง อาร์เอฟไอดีแท็ก และ บาร์โค้ด 100 ชิ้น

ประเภท	ครั้งที่					เฉลี่ย
	1	2	3	4	5	
อาร์เอฟไอดี	26.42	25.43	22.46	23.34	22.94	24.12
บาร์โค้ด	142.35	139.24	146.49	143.49	141.92	142.70

จากการทดสอบประสิทธิภาพความเร็วในการอ่านข้อมูลระหว่าง อาร์เอฟไอดี และ บาร์โค้ด ในปริมาณสินค้าที่แตกต่างกัน ตั้งแต่ 5 ชิ้น ไปจนถึง 100 ชิ้น พบประเด็นสำคัญดังนี้ ความรวดเร็วที่มากกว่าอย่างเห็นได้ชัด ในทุกจำนวนที่ทดสอบ อาร์เอฟไอดี ทำเวลาได้น้อยกว่าบาร์โค้ด โดยเฉลี่ยแล้ว ประมาณ 6-12 เท่า

ประสิทธิภาพเมื่อปริมาณสินค้าเพิ่มขึ้นที่ 5-10 ชิ้น ความแตกต่างอยู่ในระดับหลักวินาที (อาร์เอฟไอดี ใช้เวลาประมาณ 0.75 – 0.92 วินาที ในขณะที่บาร์โค้ดใช้ถึง 7.24 – 9.34 วินาที) ที่ 100 ชิ้น (ตารางที่ 4.5 จุดที่เห็นผลชัดที่สุด): อาร์เอฟไอดี ใช้เวลาเฉลี่ยเพียง 24.12 วินาที ในขณะที่บาร์โค้ดต้องใช้เวลาสูงถึง 142.70 วินาที

ในด้านอัตราการเติบโตของเวลาบาร์โค้ดมีแนวโน้มการใช้เวลาที่เพิ่มขึ้นแบบเส้นตรง (Linear) ตามจำนวนชิ้น เพราะต้องสแกนทีละชิ้น แต่ อาร์เอฟไอดี แสดงให้เห็นถึงความสามารถในการอ่านแบบกลุ่ม (Bulk Reading) ที่มีประสิทธิภาพมากกว่า แม้จำนวนชิ้นจะมากขึ้น แต่เวลาที่เพิ่มขึ้นไม่ได้สูงเท่าบาร์โค้ด

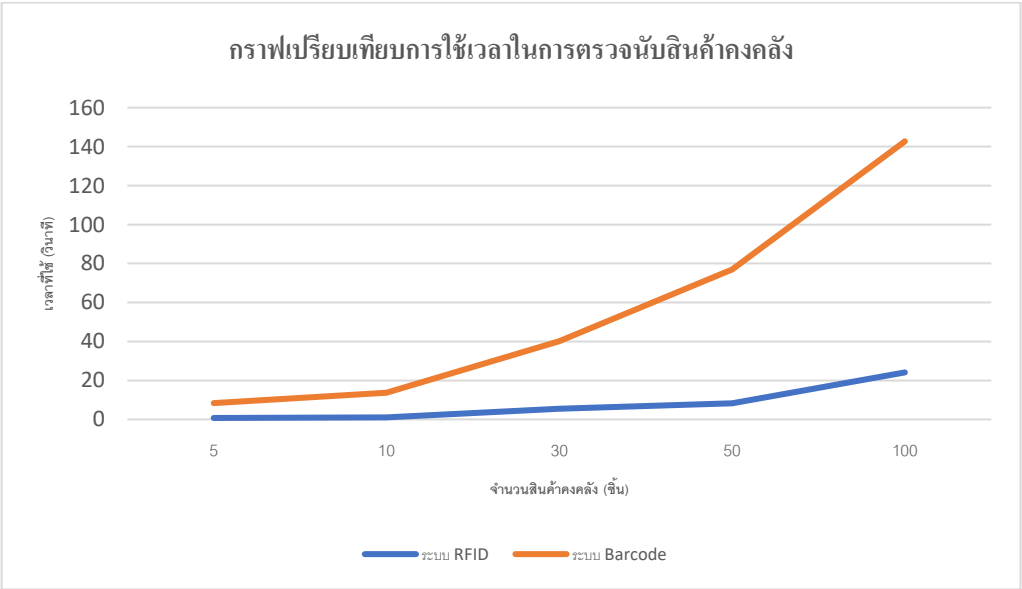
ในส่วนของความแม่นยำในการอ่านข้อมูล ได้ใช้เทคนิคการอ่านซ้ำแบบต่อเนื่อง (Continuous Sampling) เนื่องจากโมดูล อาร์เอฟไอดี มีความเร็วในการอ่านสูงมากกว่า 50 ครั้งต่อวินาที ระบบจึงถูกออกแบบให้ทำการตรวจสอบความถูกต้องของรหัสที่อ่านได้หลายครั้งภายในเสี้ยววินาที เพื่อยืนยันความถูกต้องของข้อมูลก่อนส่งไปยังฐานข้อมูล จากการทดสอบในสภาวะควบคุมพบว่าเทคนิคการอ่านซ้ำดังกล่าวช่วยลดอัตราการอ่านพลาดให้เป็นศูนย์ ส่งผลให้ระบบมีความแม่นยำในการระบุตัวตนสินค้าครบถ้วนทุกรายการ ตลอดการทดสอบทั้ง 5 ชุด ทั้งนี้ ความแม่นยำดังกล่าวขึ้นอยู่กับมุมในการอ่าน และการเรียงสินค้าซึ่งในการทดลองนี้ได้เลือกใช้ตำแหน่งที่เหมาะสมที่สุด เมื่อเทียบกับระบบที่ใช้บาร์โค้ด มีความเสี่ยงที่เกิดจากความผิดพลาดของมนุษย์ เช่น การข้ามลำดับหรือการสแกนซ้ำซ้อน แต่ระบบ อาร์เอฟไอดี ที่พัฒนาขึ้นนี้ใช้เทคนิคการกรองข้อมูลร่วมกับการอ่านซ้ำด้วยความเร็วสูง ทำให้สามารถตัดโอกาสการเกิดข้อผิดพลาดดังกล่าวได้

เนื่องจากหลักการทำงานของอาร์เอฟไอดี คือใช้คลื่นวิทยุไม่ได้ใช้แสงในการอ่านรหัส เหมือนกับ บาร์โค้ด และ คิวอาร์โค้ด เมื่อทำการเปลี่ยนบรรจุภัณฑ์ต่างชนิดกันระหว่างกล่องกระดาษ และกล่องพลาสติก ได้ผลลัพธ์ที่ใกล้เคียงกันทั้ง 2 ประเภท ดังตารางที่ 4.6

ตารางที่ 4.6 ความเร็วในการอ่านรหัสสินค้าคงคลังเมื่ออยู่ในบรรจุภัณฑ์ 2 ประเภทได้แก่กล่องกระดาษ และกล่องพลาสติก มีหน่วยเป็นวินาที

ประเภท	ครั้งที่					เฉลี่ย
	1	2	3	4	5	
กล่องกระดาษ	0.92	0.71	0.76	0.85	0.79	0.81
กล่องพลาสติก	0.84	0.62	0.78	0.87	0.68	0.76

ถึงแม้ว่าการนำระบบ อาร์เอฟไอดี มาประยุกต์ใช้งานจะมีต้นทุนด้านอุปกรณ์เริ่มต้นที่สูงกว่าระบบบาร์โค้ด และ คิวอาร์โค้ด แต่เมื่อพิจารณาเวลาในการใช้งานที่ลดลงและความแม่นยำในการอ่านรหัสจำนวนมาก และไม่ต้องสิ้นเปลืองกับการเสียเวลาและทรัพยากรไปในกรณีที่รหัสมีการเสียหายชำรุด และเลือนไปในเวลาที่เร็วกว่า อาร์เอฟไอดี Tag ระบบ อาร์เอฟไอดี จึงให้ความคุ้มค่าที่สูงมากกว่า ดังที่แสดงในภาพที่ 4.1



ภาพที่ 4.1 แสดงการเปรียบเทียบเวลาที่ใช้ในการตรวจนับสินค้า คงคลัง ระหว่าง RFID Tag และบาร์โค้ด

บทที่ 5

สรุปผลการทดลอง

การพัฒนาระบบบริหารจัดการสินค้าคงคลังด้วย อาร์เอฟไอดี และเทคโนโลยีคลาวด์ ช่วยแก้ปัญหาความล่าช้าและความยุ่งยากของระบบบาร์โค้ด และ คิวอาร์โค้ด ในระบบดั้งเดิมได้อย่างมีประสิทธิภาพ จากผลการทดลอง ยืนยันว่า การนำอาร์เอฟไอดีแท็กมาใช้แทนบาร์โค้ดช่วยลดเวลาในการตรวจสอบสินค้าคงคลังได้อย่างมีนัยสำคัญโดยเฉพาะในการบริหารจัดการสินค้าจำนวนมาก อาร์เอฟไอดีแท็กใช้เวลาน้อยกว่าบาร์โค้ดอย่างเห็นได้ชัดเจน ช่วยในการประหยัดเวลาในการทำงานและลดการสูญเสียของทรัพยากรไปกับรหัสสินค้าที่เสียหาย ชำรุด และ ลบเลือน

อีกทั้งการใช้ระบบคลาวด์ยังช่วยให้ระบบข้อมูลมีความปลอดภัยและเข้าถึงได้ง่าย ลดภาระค่าใช้จ่ายในการดูแลเครื่องแม่ข่าย และการใช้พลังงานไฟฟ้าภายในองค์กรได้อีกด้วย แต่ยังมีข้อจำกัดทางกายภาพที่ต้องคำนึงถึงในการใช้ อาร์เอฟไอดี ในการบริหารสินค้าคงคลัง เนื่องจากลักษณะการทำงานที่เป็นคลื่นวิทยุ เมื่ออยู่ใกล้วัตถุที่เป็นโลหะหรือมีของเหลว จะทำให้ประสิทธิภาพในการสื่อสารระหว่างชุดอ่านและตัว อาร์เอฟไอดีแท็ก ลดลงได้ และการจัดเรียงสินค้าในบรรจุภัณฑ์ ต้องมีความเหมาะสม ไม่ชิดกันแน่นจนทำให้เกิดการรบกวนสัญญาณระหว่าง อาร์เอฟไอดีแท็กแต่ละดวงจนเกินไป

ข้อเสนอสำหรับการพัฒนาระบบในอนาคต เพื่อให้ระบบบริหารจัดการสินค้าคงคลังรองรับการใช้งานในระดับองค์กรใหญ่ ผู้วิจัยมีข้อเสนอแนะดังนี้

1.ปรับปรุงเสาอากาศจากการใช้สายอากาศร่วมกัน (Monostatic) ไปเป็นสายอากาศแบบแยก (Bistatic) เพื่อลดปัญหาคลื่นส่งไหลกลับเข้าเครื่องรับ (Transmit Leakage) ซึ่งจะช่วยเพิ่มความไว (Sensitivity) ให้กับตัวรับสัญญาณ จะทำให้เพิ่มระยะการอ่านรหัสให้ไกลขึ้น

2.พัฒนาในส่วนของโปรแกรมให้สามารถแสดงสถานะของระบบได้ทันที เช่นการใช้ Django Channels เพื่อการแสดงผลลัพธ์เป็นปัจจุบัน

เอกสารอ้างอิง

[1] R. Neha Mukund, S. Arun, S. M. Kusuma, K. Vishak and M. Monisha, “Intelligent RFID Based Library Management System,” IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), pp. 01-06, 2021.

[2] D. Sinha, K. Cottur, K. B. H., G. C. and B. N. K., “Automated Billing System using RFID and Cloud,” Innovations in Power and Advanced Computing Technologies (i-PACT), pp. 1-6, 2019.

[3] T. D. McAllister, S. El-Tawab and M. H. Heydari, “Localization of Health Center Assets Through an IoT Environment (LoCATE),” Systems and Information Engineering Design Symposium (SIEDS), pp. 132-137, 2017.

[4] M. A. I. M. Azri, H. Sofian, M. S. M. Kassim and R. Ramly, "Smart Inventory Management System for Hardware Warehouse Using IoT," 2026 20th International Conference on Ubiquitous Information Management and Communication (IMCOM), pp. 1-6, 2026.

[5] A. P. a. others, AWS Cloud Computing Concepts and Tech Analogies, Packt Publishing, 2023.

[6] D. M. Dobkin, The RF in RFID Passive UHF RFID in practice, Elsevier Inc., 2008.

[7] J. Gale, Raspberry Pi THE COMPLETE GUIDE, Black Dog Media Limited, 2020.

[8] D. Martinez, D. Gonzalez, J. M. Sánchez and R. M. Toasa, “Library in Django framework to standardize early-stage web application development,” Iberian Conference on Information Systems and Technologies (CISTI), pp. 1-4, 2013.

[9] F. Raheel, L. Raheel, H. Mehmood, M. B. Kadri and U. S. Khan, "Internet of Robots: An Open-source Framework for Cloud-Based Global Remote Control and Monitoring of Robots," 2024 International Conference on Robotics and Automation in Industry (ICRAI), pp. 1-6, 2024.

ภาคผนวก

ภาคผนวก ก

รหัสเทียมของการตรวจสอบรหัสสินค้าด้วยบาร์โค้ด

```
คลาส BarcodeCounter:
    ฟังก์ชัน เริ่มต้น(target_count):
        กำหนด จำนวนที่ต้องการ (target_count) = target_count
        กำหนด จำนวนปัจจุบัน (current_count) = 0
        กำหนด เวลาเริ่มต้น (start_time) = ค่าว่าง
        กำหนด เวลาสิ้นสุด (end_time) = ค่าว่าง
        กำหนด รายการข้อมูล (scanned_data) = รายการว่าง (Array)

    ฟังก์ชัน เริ่มการทำงาน(run):
        พิมพ์ "เตรียมพร้อมรับข้อมูลจำนวน [จำนวนที่ต้องการ] รายการ"
        พิมพ์ "เริ่มยิงบาร์โค้ดได้เลย"

        ดักจับการกดออกโปรแกรม (KeyboardInterrupt):
            ทำซ้ำในขณะที่ (จำนวนปัจจุบัน < จำนวนที่ต้องการ):
                รับค่าข้อมูล (code) จากเครื่องสแกนบาร์โค้ดหรือคีย์บอร์ด (input)

                ถ้าข้อมูลเป็นค่าว่าง:
                    ข้ามไปรอรับข้อมูลรอบถัดไป (continue)

                ถ้าเป็นการสแกนครั้งแรก (จำนวนปัจจุบัน == 0):
                    บันทึก เวลาเริ่มต้น
                    พิมพ์ "เริ่มจับเวลา"

                เพิ่มค่า จำนวนปัจจุบัน ขึ้น 1
                เพิ่มข้อมูล (code) ลงใน รายการข้อมูล (scanned_data)
                พิมพ์ ความคืบหน้า "[จำนวนปัจจุบัน/จำนวนที่ต้องการ] รับข้อมูล: code"

            # เมื่อหลุดจากลูป (สแกนครบตามจำนวนแล้ว)
            บันทึก เวลาสิ้นสุด
            เรียกใช้ฟังก์ชัน แสดงสรุปผล(show_summary)

        เมื่อมีการกดออก (Ctrl+C):
            พิมพ์ "ยกเลิกการทำงานโดยผู้ใช้"

    ฟังก์ชัน แสดงสรุปผล(show_summary):
        ถ้ามีการบันทึก เวลาเริ่มต้น และ เวลาสิ้นสุด:
            คำนวณ เวลาทั้งหมด = เวลาสิ้นสุด - เวลาเริ่มต้น
            คำนวณ ความเร็วเฉลี่ย = เวลาทั้งหมด / จำนวนที่ต้องการ
            พิมพ์ สรุปผลการทำงาน (เวลาทั้งหมดที่ใช้ และ ความเร็วเฉลี่ยต่อรายการ)
        มิฉะนั้น:
            พิมพ์ "ยังไม่มีข้อมูลเพียงพอสำหรับการคำนวณ"


ส่วนหลักของโปรแกรม (Main Program):
    กำหนด จำนวนเริ่มต้น (target_n) = 1

    ถ้ามีการรับค่า (Argument) ผ่าน Command Line มาด้วย:
        พยายามแปลงค่าที่รับมาเป็นตัวเลขจำนวนเต็ม (Integer):
            ถ้า ค่าน้อยกว่าหรือเท่ากับ 0:
                พิมพ์ "กรุณาใส่จำนวนเต็มที่มากกว่า 0"
                ออกจากโปรแกรม
        ถ้าแปลงเป็นตัวเลขไม่ได้:
            พิมพ์ "ไม่ใช่ตัวเลข กรุณาใส่จำนวนเต็ม"
            ออกจากโปรแกรม

    สร้างออบเจกต์ app จากคลาส BarcodeCounter โดยส่งค่า target_n ไป
    สั่งให้ app.เริ่มการทำงาน()
```

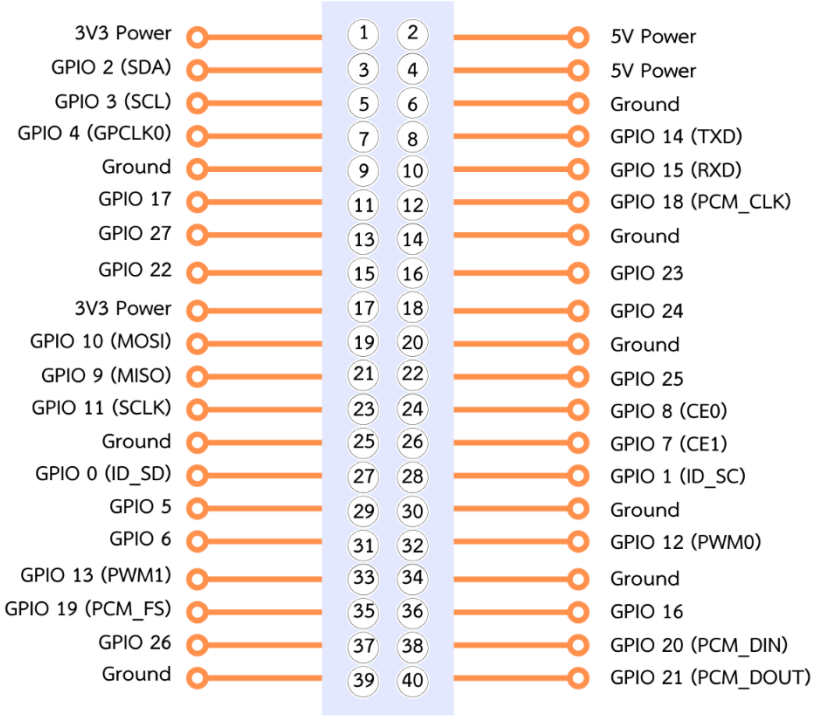
ภาคผนวก ข

รหัสเทียมของการตรวจสอบรหัสสินค้าด้วยอาร์เอฟไอดี

```
การตั้งค่าฮาร์ดแวร์ (Hardware Setup):
    เปิดใช้งานขา GPIO ควบคุมโมดูล RFID (ขา 4 เป็น HIGH)
    เชื่อมต่อ Serial Port เบื้องต้น (Baudrate 38400)

คลาส RFCounter:
    ฟังก์ชัน เริ่มต้น(target_count):
        กำหนด จำนวนเป้าหมาย = target_count
        กำหนด จำนวนปัจจุบัน = 0
        กำหนด ตัวแปรจับเวลา (เวลาเริ่ม, เวลาสิ้นสุด) = ค่าว่าง
        กำหนด รายการแท็กที่อ่านแล้ว (scanned_data) = รายการว่าง (Array)
        กำหนด จำนวนรอบการส่งคำสั่ง (count_cmd_loop) = 0

    ฟังก์ชัน เริ่มการทำงาน(run):
        พิมพ์ "เตรียมพร้อมรับข้อมูล RFID จำนวน [จำนวนเป้าหมาย] รายการ"

        ดักจับการกดออกโปรแกรม (KeyboardInterrupt):
            ทำซ้ำในขณะที่ (จำนวนปัจจุบัน < จำนวนเป้าหมาย):
                ล้าง Buffer ขาออก
                ส่งคำสั่ง "อ่านแท็กต่อเนื่อง (Multi-Query EPC / Command U)" ไปที่โมดูล
                หน่วงเวลาสั้นๆ (0.1 วินาที) เพื่อรอโมดูลอ่านแท็กและส่งกลับมา

                # ลูปอ่านข้อมูลที่โมดูลตอบกลับมา
                ทำซ้ำตลอดไป:
                    ถ้ามีข้อมูลรออยู่ใน Buffer:
                        อ่านข้อมูลมา 1 บรรทัด

                        ถ้าบรรทัดนั้นมีแค่ตัวอักษร 'U' ตัวเดียว:
                            แปลว่าโมดูลส่งข้อมูลของรอบนี้ครบแล้ว -> ให้ออกจากลูปอ่านข้อมูลนี้ (break)

                        ถ้าข้อมูลมีรหัส 'E2' (บ่งบอกว่าเป็นข้อมูล RFID รูปแบบ EPC Gen2):
                            ตัดเอาเฉพาะข้อมูล 8 หลักสุดท้ายมาเป็นตัวแปรรหัสแท็ก (code)

                            นับเพิ่ม จำนวนรอบการส่งคำสั่ง ขึ้น 1

                            ถ้าเป็นการอ่านเจอครั้งแรก (จำนวนปัจจุบัน == 0):
                                บันทึก เวลาเริ่มต้น

                            # เช็คข้อมูลซ้ำ
                            ถ้ารหัสแท็กนี้ยังไม่มีใน รายการแท็กที่อ่านแล้ว:
                                เพิ่ม จำนวนปัจจุบัน ขึ้น 1
                                บันทึกรหัสแท็กลงใน รายการแท็กที่อ่านแล้ว
                                พิมพ์แสดงความคืบหน้า "[จำนวนปัจจุบัน/จำนวนเป้าหมาย] รับข้อมูล: code"

                หน่วงเวลาสั้นๆ (0.01 วินาที) ป้องกัน Buffer เต็มก่อนเริ่มลูปรอบถัดไป

            # เมื่อหลุดลูปหลัก (อ่านครบตามจำนวนแล้ว)
            บันทึก เวลาสิ้นสุด
            เรียกใช้ฟังก์ชัน แสดงสรุปผล()

        เมื่อมีการกดออก (Ctrl+C):
            ปิดการเชื่อมต่อ Serial Port (ถ้าจำเป็น) / ยกเลิกการทำงาน

    ฟังก์ชัน แสดงสรุปผล(show_summary):
        คำนวณ เวลาทั้งหมดที่ใช้ = เวลาสิ้นสุด - เวลาเริ่มต้น
        คำนวณ ความเร็วเฉลี่ย = เวลาทั้งหมดที่ใช้ / จำนวนเป้าหมาย
        พิมพ์ สรุปผลการทำงาน (เวลาทั้งหมด, ความเร็วเฉลี่ย, และ จำนวนรอบที่ส่งคำสั่งอ่านทั้งหมด)

ส่วนหลักของโปรแกรม (Main Program):
    กำหนด จำนวนเริ่มต้น = 1

    ถ้ามีการส่งค่าตัวเลขต่อท้ายตอนรันโปรแกรม (Command Line Argument):
        พยายามแปลงค่าเป็นตัวเลขจำนวนเต็ม
            ถ้าน้อยกว่า 1 หรือไม่ใช่ตัวเลข:
                แสดงข้อความ Error และออกจากโปรแกรม
        ตั้งค่า จำนวนเป้าหมาย = ค่าที่รับมา

    สร้างออบเจกต์ app จากคลาส RFCounter (พร้อมส่ง จำนวนเป้าหมาย เข้าไป)
    สั่งให้ app.เริ่มการทำงาน()
```

ภาคผนวก ค

Raspberry Pi 5 Datasheet



ประวัติผู้เขียน

ชื่อ-นามสกุล	นาย อภิชาติ กันสีนวล
ประวัติการศึกษา	2543 วิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยมหาสารคาม
สถานที่ทำงานปัจจุบัน	นักวิชาการคอมพิวเตอร์ ศาลแขวงเชียงดาว