



รายงานปฏิบัติงานสหกิจศึกษา

เรื่อง

ระบบปลูกสตรอเบอรี่อัจฉริยะด้วยปัญญาประดิษฐ์และไอโอที
Smart strawberry growing system using artificial intelligence and IoT.

โดย

นายอลงกรณ์ ไชยสา
671540006012-2

ปฏิบัติงาน ณ

คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมอุตสาหการ
สาขาวิศวกรรมปัญญาประดิษฐ์ และการสั่งการ
เลขที่ 85 ถนนสถลมารค ตำบลศรีไค
อำเภอวารินชำราบ จังหวัดอุบลราชธานี รหัสไปรษณีย์ 34190

รายงานนี้เป็นส่วนหนึ่งของการเรียนรายวิชาสหกิจศึกษา
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม
มหาวิทยาลัยกาฬสินธุ์
ปีการศึกษา 2568



รายงานปฏิบัติงานสหกิจศึกษา

เรื่อง

ระบบปลูกสตรอเบอรี่อัจฉริยะด้วยปัญญาประดิษฐ์และไอโอที
Smart strawberry growing system using artificial intelligence and IoT.

โดย

นายอลงกรณ์ ไชยสา
671540006012-2

ปฏิบัติงาน ณ

คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมอุตสาหการ
สาขาวิศวกรรมปัญญาประดิษฐ์ และการสั่งการ
เลขที่ 85 ถนนสถลมารค ตำบลศรีไค
อำเภอวารินชำราบ จังหวัดอุบลราชธานี รหัสไปรษณีย์ 34190

รายงานนี้เป็นส่วนหนึ่งของการเรียนรายวิชาสหกิจศึกษา
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม
มหาวิทยาลัยกาฬสินธุ์
ปีการศึกษา 2568

กิตติกรรมประกาศ (Acknowledgement)

การที่ข้าพเจ้าได้มาปฏิบัติงานสหกิจศึกษา ณ มหาลัยอุบลราชธานี ตั้งแต่วันที่ 17 เดือน พฤศจิกายน พ.ศ. 2568 ถึง วันที่ 17 เดือน มีนาคม พ.ศ. 2569 ส่งผลให้ข้าพเจ้าได้รับความรู้และประสบการณ์ต่างๆ ที่มีค่ามากมาย สำหรับรายงานวิชาสหกิจศึกษานี้ สำเร็จลงได้ด้วยดีจากความร่วมมือและสนับสนุนจากหลายฝ่าย ดังนี้

1. ชื่อ-สกุล.....ตำแหน่ง.....
2. ชื่อ-สกุล.....ตำแหน่ง.....
3. ชื่อ-สกุล.....ตำแหน่ง.....
4. ชื่อ-สกุล.....ตำแหน่ง.....
5. ชื่อ-สกุล.....ตำแหน่ง.....
6. ระบุคนอื่น ๆ ที่ได้ให้คำแนะนำ ช่วยเหลือ ในการจัดทำรายงาน

นอกจากนี้ขอขอบคุณบิดา - มารดา อาจารย์ และบุคคลท่านอื่นๆ ที่ไม่ได้กล่าวนามไว้ ณ ที่นี้ ซึ่งท่านเหล่านี้ได้กรุณาให้คำแนะนำช่วยเหลือในการจัดทำรายงานฉบับนี้ ข้าพเจ้าใคร่ขอขอบพระคุณผู้ที่มีส่วนเกี่ยวข้องทุกท่านที่มีส่วนร่วมในการให้ข้อมูล คำแนะนำ และวิธีการปฏิบัติงาน รวมถึงเป็นที่ปรึกษาในการจัดทำรายงานฉบับนี้จนเสร็จสมบูรณ์

ลงชื่อ

(นายอลกรณ์ ไชยสา)

นักศึกษาผู้ปฏิบัติงานสหกิจศึกษา

วันที่.....เดือน.....พ.ศ.....

บทคัดย่อ (Abstract)

รายงานฉบับนี้รวบรวมองค์ความรู้และการออกแบบระบบวิศวกรรมสำหรับโรงงาน เรือนผลิตพืช (Plant Factory) เพื่อการปลูกสตรอเบอรี่พันธุ์พระราชทาน 80 โดยมีวัตถุประสงค์เพื่อตั้งศักยภาพทาง พันธุ์กรรมของพืชออกมาให้ได้สูงสุด ทั้งในด้านความหวาน (TSS) ความแน่นของเนื้อ และปริมาณผลผลิต โครงการนี้บูรณาการองค์ความรู้ด้านเกษตรแม่นยำเข้ากับเทคโนโลยีสมัยใหม่ ได้แก่ การควบคุมสภาวะ แวดล้อมอัตโนมัติด้วยไมโครคอนโทรลเลอร์ ESP32, การใช้ปัญญาประดิษฐ์ (AI) ด้วยโมเดล YOLOv8 ในการวิเคราะห์ภาพถ่ายเพื่อทำหน้าที่เป็น Soft Sensors และการพัฒนาแพลตฟอร์มแสดงผลข้อมูลแบบ เรียลไทม์ด้วย Server ผลลัพธ์ที่ได้คือระบบนิเวศการปลูกที่มีความแม่นยำสูง สามารถตอบสนองต่อความ ต้องการของพืชได้อย่างทันท่วงที และลดความเสี่ยงจากการตัดสินใจโดยอาศัยเพียงประสบการณ์ของผู้ปลูก

ข้าพเจ้าได้รับมอบหมายให้ปฏิบัติงานในตำแหน่ง.....(ตำแหน่งที่ได้รับมอบหมาย)..... บริษัท.....จำกัด ซึ่งมี.....(ชื่อผู้มอบหมายงาน)..... เป็นผู้มอบหมายงาน โดยงานที่ ข้าพเจ้าได้รับมอบหมายเกี่ยวข้องกับ..... เช่น..... เป็นต้น จากการปฏิบัติงานดังกล่าวข้างต้นทำให้ข้าพเจ้าได้รับประโยชน์มากมายไม่ว่าจะเป็นการวางแผน ก่อนการทำงาน การบริหารจัดการเวลาที่มีอยู่อย่างจำกัด การปรับตัวให้กับผู้อื่น การค้นพบข้อบกพร่องของ ตนเองในด้านความรู้และทักษะการทำงาน ซึ่งสิ่งต่างๆ เหล่านี้เป็นบทเรียนแรกที่ใช้เป็นแนวทางในการพัฒนา ด้านการทำงานในอนาคตของข้าพเจ้า

คำสำคัญ : สหกิจศึกษา/ระบบปลูกสตรอเบอรี่อัจฉริยะด้วยปัญญาประดิษฐ์และไอโอที

สารบัญ

| | |
|--|----|
| สารบัญภาพ | ก |
| สารบัญตาราง | ข |
| บทที่ 1 บทนำ | 3 |
| 1.1 ชื่อและที่ตั้งสถานประกอบการ | 3 |
| 1.2 ลักษณะการประกอบการ ผลิตภัณฑ์ การให้บริการหลักขององค์กร | 3 |
| 1.2.1 การวิจัยและพัฒนาระบบอัตโนมัติ | 3 |
| 1.2.2 การพัฒนาปัญญาประดิษฐ์ | 3 |
| 1.2.3 อินเทอร์เน็ตของสรรพสิ่ง | 3 |
| 1.3.1 คณะกรรมการบริหารภาควิชา | 3 |
| 1.3.2 หน่วยวิจัยเฉพาะทาง (Research Unit) | 3 |
| 1.3.3 หน่วยสนับสนุนทางวิศวกรรม | 3 |
| บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง | 5 |
| 2.1 .ความเป็นมาและความสำคัญของปัญหา | 5 |
| 2.6 อุปกรณ์และเครื่องมือที่ใช้ | 7 |
| 2.6.1 คอมพิวเตอร์ (Computer) | 7 |
| 2.6.2 ไมโครคอนโทรลเลอร์ ESP32 (Processing Unit) | 9 |
| 2.6.3 โมดูลเซ็นเซอร์วัดอุณหภูมิและความชื้น (DHT22) | 10 |
| 2.6.4 หัววัดค่าความเป็นกรด-ด่าง (pH Probe) | 13 |
| 2.6.5 เซ็นเซอร์วัดค่าการนำไฟฟ้า (TDS Sensor) | 15 |
| 2.6.7.1 ภาษาไพธอน (Python Programming Language) | 19 |
| 2.6.8 โปรแกรมวิซวลสตูดิโอโค้ด (Visual Studio Code) | 21 |
| 2.7 รายละเอียดของโครงการ | 23 |
| 3.1 สิ่งที่คาดหวัง | 24 |
| 3.3 วิเคราะห์จุดเด่น จุดด้อย โอกาส อุปสรรค (Swot Analysis) | 24 |

สารบัญภาพ

| | |
|--|----|
| ภาพที่ 1 ตำแหน่งงานของพนักงานที่ปรึกษา | 4 |
| ภาพที่ 2 คอมพิวเตอร์อิเล็กทรอนิกส์ | 7 |
| ภาพที่ 3 ไมโครคอนโทรลเลอร์ ESP32 | 9 |
| ภาพที่ 4 โมดูลเซ็นเซอร์วัดอุณหภูมิและความชื้น | 12 |
| ภาพที่ 5 เซ็นเซอร์เคมีไฟฟ้า pH Probe | 13 |
| ภาพที่ 6 สูตรคำนวณค่า pH Probe | 14 |
| ภาพที่ 7 เซ็นเซอร์วัดค่าการนำไฟฟ้า TDS Sensor | 15 |
| ภาพที่ 8 เฟรมเวิร์กสำหรับการพัฒนาเว็บแอปพลิเคชัน | 17 |
| ภาพที่ 9 การทำงานของสถาปัตยกรรมหลัก | 17 |
| ภาพที่ 10 ภาษาโปรแกรม Python | 19 |
| ภาพที่ 11 โปรแกรมแก้ไขซอร์สโค้ด VS Code | 21 |

สารบัญตาราง

| | |
|---|----|
| ตารางที่ 1 เปรียบเทียบอุปกรณ์ ESP32 และ ESP8266 | 10 |
| ตารางที่ 2 คุณสมบัติ DHT22 | 11 |
| ตารางที่ 3 คุณสมบัติ Parameter | 14 |
| ตารางที่ 4 ความแตกต่าง React และ Next.js | 19 |

บทที่ 1

บทนำ

1.1 ชื่อและที่ตั้งสถานประกอบการ

มหาวิทยาลัยอุบลราชธานี คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมอุตสาหการ สาขาวิศวกรรมปัญญาประดิษฐ์ และการจัดการ กลุ่มวิจัยการหาค่าที่ดีที่สุดในระบบปัญญาประดิษฐ์สำหรับเมืองอัจฉริยะ (Artificial Intelligence Optimization for Elderly-Friendly Smart City หรือ AIO-Smart Lab)

เลขที่ 85 ถนนสถลมารค ตำบลศรีโค อำเภวารินชำราบ จังหวัดอุบลราชธานี รหัสไปรษณีย์ 34190

1.2 ลักษณะการประกอบการ ผลิตภัณฑ์ การให้บริการหลักขององค์กร

ภาควิชาวิศวกรรมอุตสาหการ มหาวิทยาลัยอุบลราชธานี โดยเฉพาะในหลักสูตรวิศวกรรมปัญญาประดิษฐ์และการจัดการ เป็นหน่วยงานที่มุ่งเน้นการผลิตนวัตกรรมและงานวิจัยเชิงวิศวกรรมขั้นสูง การดำเนินงานหลักที่เกี่ยวข้องกับโครงการนี้ประกอบด้วย

1.2.1 การวิจัยและพัฒนาระบบอัตโนมัติ (Automation R&D) การออกแบบระบบควบคุมที่ใช้ในอุตสาหกรรมและเกษตรกรรมสมัยใหม่

1.2.2 การพัฒนาปัญญาประดิษฐ์ (AI Development) การนำ Deep Learning และ Computer Vision มาใช้ในการแก้ปัญหาเฉพาะทาง เช่น การตรวจสอบคุณภาพผลผลิตทางการเกษตร

1.2.3 อินเทอร์เน็ตของสรรพสิ่ง (IoT) การเชื่อมต่ออุปกรณ์สมองกลฝังตัวเข้ากับเครือข่ายเพื่อการจัดการข้อมูลขนาดใหญ่ (Big Data) ในพื้นที่การเกษตรแม่นยำ (Precision Farming)

1.3 รูปแบบการจัดองค์การและการบริหารงานขององค์กร

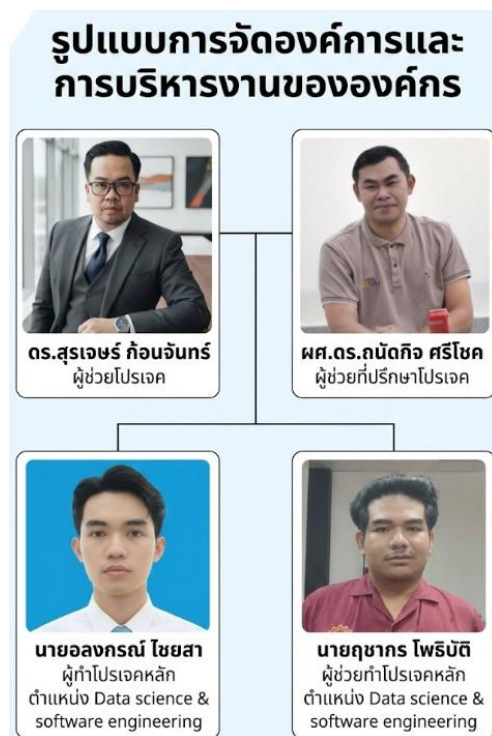
หน่วยงานมีการบริหารงานภายใต้กำกับของคณะวิศวกรรมศาสตร์ มหาวิทยาลัยอุบลราชธานี โดยมีการแบ่งส่วนงานที่สนับสนุนโครงการวิจัยดังนี้

1.3.1 คณะกรรมการบริหารภาควิชา กำหนดนโยบายและทิศทางการวิจัยให้สอดคล้องกับยุทธศาสตร์ชาติด้านเกษตรอัจฉริยะ

1.3.2 หน่วยวิจัยเฉพาะทาง (Research Unit) รับผิดชอบการทดลองเรื่องสเปกตรัมแสงและระดับสารอาหารที่เหมาะสมสำหรับพืชเมืองหนาว (พันธุ์พระราชทาน 80) ในระบบปิด

1.3.3 หน่วยสนับสนุนทางวิศวกรรม ดูแลอุปกรณ์เครื่องมือวัดระดับอุตสาหกรรม (Industrial Sensors) และโครงสร้างพื้นฐานด้านเครือข่ายคอมพิวเตอร์

1.4 ตำแหน่งและลักษณะงานที่ได้รับมอบหมาย



ภาพที่ 1 ตำแหน่งงานของพนักงานที่ปรึกษา

1.5 ชื่อและตำแหน่งงานของพนักงานที่ปรึกษา

ดร.สุรเชษฐ์ ก้อนจันทร์ ผู้ช่วยโปรเจกต์

ผศ.ดร.ณัฏกิจ ศรีโชค ผู้ช่วยที่ปรึกษาโปรเจกต์

นายอลงกรณ์ ไชยสา ผู้ทำโปรเจกต์หลัก ตำแหน่ง วิศวกรรมซอฟต์แวร์

นายฤชากร โพธิ์บัติ ผู้ช่วยทำโปรเจกต์หลัก ตำแหน่ง วิศวกรรมซอฟต์แวร์

1.6 ระยะเวลาที่ปฏิบัติงาน

ปีการศึกษา 2568 ตั้งแต่วันที่ 17 พฤศจิกายน 2568 ถึงวันที่ 17 มีนาคม 2569 รวมเป็นเวลา 18 สัปดาห์

บทที่ 2

รายละเอียดการปฏิบัติงาน

2.1 . ความเป็นมาและความสำคัญของปัญหา

สตอเบอร์รี่พันธุ์พระราชทาน 80 เป็นพืชเศรษฐกิจที่มีมูลค่าสูงแต่มีข้อจำกัดด้านสภาพภูมิอากาศที่ต้องพึ่งพาความหนาวเย็นและการจัดการธาตุอาหารที่แม่นยำ การปลูกในระบบเปิดมักประสบปัญหาความไม่แน่นอนของผลผลิต คุณภาพความหวาน (TSS) ที่ไม่สม่ำเสมอ และความเสี่ยงต่อโรคพืช

ในปัจจุบันเทคโนโลยี Plant Factory with Artificial Lighting (PFAL) เริ่มเข้ามามีบทบาท แต่การปลูกในระบบปิดยังเผชิญกับปัญหาการสะสมของความร้อนจากแสงไฟ LED และความชื้นสัมพัทธ์ที่อาจก่อให้เกิดอาการใบไหม้ (Tip burn) หากไม่มีระบบระบายอากาศที่มีประสิทธิภาพเพียงพอ นอกจากนี้ การตัดสินใจปรับค่าสภาวะแวดล้อมมักอาศัยประสบการณ์ของผู้ปลูกเป็นหลัก ซึ่งมีความเสี่ยงต่อความผิดพลาดสูง ดังนั้น การนำระบบ IoT มาใช้ร่วมกับปัญญาประดิษฐ์ (YOLOv8) เพื่อทำหน้าที่เป็น "Soft Sensors" วิเคราะห์สุขภาพพืชและระยะการสุกแบบ Real-time จึงเป็นแนวทางสำคัญในการเพิ่มศักยภาพการผลิตให้ได้คุณภาพตามมาตรฐานสูงสุดสม่ำเสมอตลอดทั้งปี

2.2 วัตถุประสงค์การปฏิบัติงาน

2.2.1 เพื่อออกแบบและสร้างระบบควบคุมสภาวะแวดล้อมอัตโนมัติ (IoT) สำหรับการปลูกสตอเบอร์รี่พันธุ์พระราชทาน 80 ในระบบปิด

2.2.2 เพื่อพัฒนาโมเดลปัญญาประดิษฐ์ (YOLOv8) ในการตรวจจับความสุกและโรคพืชจากภาพถ่าย

2.2.3 เพื่อวิจัยและทดสอบสเปกตรัมแสง (Ratio 1:6) และระดับธาตุอาหารที่เหมาะสมต่อความหวานและผลผลิต

2.2.4 เพื่อพัฒนา Web Dashboard (Next.js) สำหรับการติดตามข้อมูลและสั่งการควบคุมทางไกล

2.2.5 เพื่อสร้างระบบวงจรป้อนกลับ (Feedback Loop) ระหว่างปัญญาประดิษฐ์และฮาร์ดแวร์เพื่อการตัดสินใจที่แม่นยำ

2.3 ขอบเขตของโครงการ

2.3.1 ศึกษาและทดลองกับสตอเบอร์รี่พันธุ์พระราชทาน 80 เท่านั้น

2.3.2 ระบบควบคุมฮาร์ดแวร์ใช้ไมโครคอนโทรลเลอร์ ESP32 เป็นตัวประมวลผลหลัก

2.3.3 การวิเคราะห์ภาพใช้โมเดล YOLOv8n (Nano) เพื่อการประมวลผลที่รวดเร็ว (Real-time)

2.3.4 ระบบสื่อสารข้อมูลใช้โปรโตคอล MQTT สำหรับการควบคุม และ Firebase RTDB สำหรับการจัดเก็บข้อมูลเซ็นเซอร์

2.3.5 ขอบเขตพื้นที่การปลูกจำกัดอยู่ในกล่องโฝมระบบปิด (Plant Factory) ขนาดที่กำหนด

2.4 ประโยชน์ที่คาดว่าจะได้รับ

- 2.4.1 สามารถควบคุมสภาวะแวดล้อมให้เหมาะสมกับพันธุกรรมของพืชเพื่อผลิตสูงสุด
- 2.4.2 ลดการสูญเสียของผลผลิตจากการตรวจพบโรคพืชและอาการใบไหม้ (Tip burn) ได้อย่างทันทั่วทั้งผ่าน AI
- 2.4.3 ได้ระบบปลูกที่สามารถเพิ่มค่าความหวาน (TSS) ของสตรอเบอร์รี่ผ่านการปรับค่า EC อัตโนมัติก่อนการเก็บเกี่ยว
- 2.4.4 ผู้ปลูกสามารถบริหารจัดการฟาร์มได้จากระยะไกล ลดภาระงานและการใช้แรงงานคน
- 2.4.5 เป็นต้นแบบงานวิจัยด้านเกษตรแม่นยำ (Precision Agriculture) สำหรับพืชเมืองหนาว

2.5 ขั้นตอนและวิธีการดำเนินงาน

- 2.5.1 การศึกษาข้อมูล รวบรวมค่ามาตรฐาน PPFD
- 2.5.2 การออกแบบระบบ เขียนผังการไหลของข้อมูล (Data Flow) และสถาปัตยกรรมระบบรวม (System Architecture)
- 2.5.3 การพัฒนาฮาร์ดแวร์ ประกอบวงจร ESP32 เชื่อมต่อเซ็นเซอร์ DHT22, DS18B20 และชุดควบคุม Relay
- 2.5.4 การพัฒนา AI รวบรวมข้อมูลภาพสตรอเบอร์รี่ ทำ Data Labeling และ Train โมเดล YOLOv8
- 2.5.5 การพัฒนาระบบซอฟต์แวร์ พัฒนา Frontend ด้วย Next.js และเขียนระบบ Backend ด้วย Python สำหรับประมวลผล AI
- 2.5.6 การทดสอบและปรับปรุง ทดสอบการทำงานร่วมกัน (Integration Test) และวัดผลค่าความหวานสตรอเบอร์รี่จริง

2.6 อุปกรณ์และเครื่องมือที่ใช้



ภาพที่ 2 คอมพิวเตอร์อิเล็กทรอนิกส์

2.6.1 คอมพิวเตอร์ (Computer)

คอมพิวเตอร์ คือ เครื่องมือหรืออุปกรณ์คอมพิวเตอร์ computer นิยมอ่านในภาษาไทยว่า คอมพิวเตอร์อิเล็กทรอนิกส์ ที่มีความสามารถในการคำนวณอัตโนมัติตามคำสั่ง ส่วนที่ใช้ประมวลผลเรียกว่าหน่วยประมวลผลชุดของคำสั่งที่ระบุขั้นตอนการคำนวณ เรียกว่าโปรแกรมคอมพิวเตอร์ ผลลัพธ์ที่ได้ออกมานี้อาจเป็นได้ทั้ง ตัวเลข ข้อความ รูปภาพ เสียง หรืออยู่ในรูปอื่น ๆ อีกมากมาย

ลักษณะทางกายภาพของคอมพิวเตอร์นั้นมีหลากหลาย มีทั้งขนาดที่ใหญ่มากจนต้องใช้ห้องทั้งห้องในการบรรจุ และขนาดเล็กจนวางได้บนฝ่ามือ การจัดแบ่งประเภทของคอมพิวเตอร์สามารถจัดแบ่งได้ตามขนาดทางกายภาพเป็นสำคัญ ซึ่งมักจะแปรผันกับประสิทธิภาพความเร็วในการประมวลผลโดยขนาดคอมพิวเตอร์ ที่มีขนาดใหญ่ที่สุดเรียกว่า ซูเปอร์คอมพิวเตอร์ ใช้กับการคำนวณผลทางวิทยาศาสตร์ ขนาดรองลงมาเรียกว่า เมนเฟรม มักใช้ในบริษัทขนาดใหญ่ที่ต้องมีการประมวลผลธุรกรรมทางธุรกิจจำนวนมากๆ สำหรับคอมพิวเตอร์ขนาดเล็กที่ใช้ในระดับบุคคลเรียกว่า คอมพิวเตอร์ส่วนบุคคล และคอมพิวเตอร์ส่วนบุคคลที่พกพาได้เรียกว่า คอมพิวเตอร์โน้ตบุ๊ก ส่วนคอมพิวเตอร์ขนาดเล็กที่สามารถวางบนฝ่ามือได้เรียกว่า พีดีเอ อย่างไรก็ตามคอมพิวเตอร์มีใช้กันอย่างกว้างขวางมาก ซึ่งมีอุปกรณ์หลายๆชนิดได้นำคอมพิวเตอร์ ไปใช้เป็นกลไกหลักในการทำงาน เช่น กล้องดิจิทัล เครื่องเล่นเอ็มพีสามหรือในรถยนต์เองก็มีคอมพิวเตอร์ที่ใช้ช่วยในการตรวจสอบระบบการทำงานของเครื่องยนต์

เครื่องคอมพิวเตอร์ทำหน้าที่เป็น AI Server หลักที่คอยรับข้อมูลภาพจาก ESP32-CAM มาประมวลผลด้วยอัลกอริทึม YOLOv8 โดยมีรายละเอียดองค์ประกอบและการทำงานดังนี้

2.6.1.1 องค์ประกอบสำคัญของคอมพิวเตอร์ในโครงงาน (Key Components) เพื่อให้ AI ประมวลผลได้แบบ Real-time องค์ประกอบของคอมพิวเตอร์ที่ใช้ในงานนี้มีหน้าที่เฉพาะเจาะจงดังนี้

ฮาร์ดแวร์ (Hardware)

2.6.1.2 CPU (Central Processing Unit) เปรียบเสมือนสมองสั่งการ จัดการการรับส่งข้อมูลผ่านโปรโตคอล MQTT และจัดการคิวของข้อมูลเซ็นเซอร์

2.6.1.3 GPU (Graphics Processing Unit) สำคัญมากสำหรับโครงงานนี้ใช้ประมวลผลโครงข่ายประสาทเทียม (Neural Network) ของ YOLOv8 เพื่อให้สามารถตรวจจับความสุกของสตรอเบอร์รี่ได้อย่างรวดเร็ว

2.6.1.3 RAM หน่วยความจำสำรองที่ใช้พักข้อมูลภาพชั่วคราวก่อนส่งให้ AI วิเคราะห์

ซอฟต์แวร์ (Software)

2.6.1.3 System Software เช่น Windows 11 หรือ Ubuntu Linux ที่ป็นฐานปฏิบัติการ

Application Software: ได้แก่ Python (ภาษาหลัก), OpenCV (จัดการภาพ), และ Next.js (ส่วนหน้าจอบทความ)

2.6.1.4 ข้อมูล (Data) ภาพถ่ายสตรอเบอร์รี่จากกล้อง และค่าตัวเลขจากเซ็นเซอร์ (pH, EC, Temp)

2.6.1.5 บุคลากร (Peopleware) ผู้พัฒนา (ตัวคุณ) ที่เขียนโปรแกรมควบคุม และผู้ใช้งานที่ดูข้อมูลผ่าน Dashboard

วงจรการทำงานของคอมพิวเตอร์ (Computer Workflow) การทำงานของคอมพิวเตอร์ในระบบปลูกสตรอเบอร์รี่อัจฉริยะนี้ แบ่งออกเป็น 4 ขั้นตอนตามหลักการพื้นฐาน

1. การรับข้อมูล (Input) คอมพิวเตอร์รับ "สัญญาณภาพ" จาก ESP32-CAM ผ่านระบบเครือข่ายไร้สาย (Wi-Fi) และรับ "ค่าสถานะแวดล้อม" (เช่น อุณหภูมิ) จาก ESP32 ผ่านโปรโตคอล MQTT

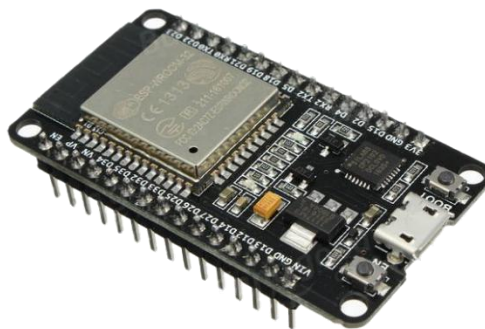
2. การประมวลผล (Processing) นี่คือนาทีที่คอมพิวเตอร์ใช้ตรรกะและคณิตศาสตร์ขั้นสูง

นำภาพเข้าสู่โมเดล YOLOv8 เพื่อจำแนกวัตถุ คำนวณสถิติ: เช่น "ถ้าพื้นที่สีแดงบนผลสตรอเบอร์รี่ > 70\% ให้ถือว่าสุก"

Logical Logic: $\text{IF Ripeness} > 70\% \text{ THEN Send Command to Pump}$

3. การจัดเก็บข้อมูล (Storage) บันทึกข้อมูลภาพและค่าที่วัดได้ลงใน Database (Firebase) เพื่อนำไปสร้างกราฟวิเคราะห์ย้อนหลังว่าในช่วงเดือนที่ผ่านมา สตรอเบอร์รี่มีการเติบโตอย่างไร

4. การแสดงผล (Output) ส่งผลลัพธ์ที่ประมวลผลเสร็จแล้วไปแสดงที่ Web Dashboard เพื่อให้ผู้ปลูกทราบสถานะ และส่งคำสั่งควบคุมกลับไปยังตัว Hardware (Relay)



ภาพที่ 3 ไมโครคอนโทรลเลอร์ ESP32

2.6.2 ไมโครคอนโทรลเลอร์ ESP32 (Processing Unit)

ESP32 เป็นโมดูลไมโครคอนโทรลเลอร์ที่เหมาะสมที่สุดสำหรับงาน Precision Agriculture (เกษตรแม่นยำ) ในโครงการนี้ เนื่องจากมีคุณสมบัติที่โดดเด่นดังนี้

2.6.2.1 สถาปัตยกรรมแบบ Dual-Core ความสำคัญ ESP32 มี CPU สองคอร์ (Core 0 และ Core 1) ทำให้เราสามารถแยกการทำงานได้ เช่น ให้คอร์หนึ่งจัดการการเชื่อมต่อ WiFi/MQTT และอีกคอร์หนึ่งทำหน้าที่อ่านค่าจาก pH/EC Sensor และควบคุม Relay แบบ Real-time วิธีนี้ช่วยป้องกันการค้างของระบบ (System Freeze) เมื่อสัญญาณอินเทอร์เน็ตรั่วมีปัญหา

2.6.2.2 ภาคการแปลงสัญญาณ (ADC & DAC) ความแม่นยำ ESP32 มาพร้อมกับ ADC (Analog to Digital Converter) ขนาด 12-bit (ให้ค่า 0-4095) ซึ่งละเอียดกว่า Arduino ทั่วไป (10-bit) ทำให้สามารถอ่านค่าความเข้มข้นของปุ๋ย (EC) และค่าความเป็นกรด-ด่าง (pH) ได้ละเอียดและแม่นยำมากขึ้น

2.6.1.3 การประหยัดพลังงาน (Low Power Management) Deep Sleep แม้ในระบบ Plant Factory จะเสียไฟตลอดเวลา แต่ความสามารถในการเข้าโหมด Deep Sleep ช่วยลดความร้อนสะสมในตัวชิป ซึ่งอาจส่งผลต่อการอ่านค่าอุณหภูมิที่คลาดเคลื่อนได้

ตารางที่ 1 เปรียบเทียบอุปกรณ์ ESP32 และ ESP8266

| คุณสมบัติ | ESP8266 | ESP32 | ประโยชน์ต่อโครงงานสตรอเบอร์รี่ |
|------------------|---------------|------------------|---|
| จำนวน Core | Single Core | Dual Core | ประมวลผลเซ็นเซอร์พร้อมส่งข้อมูลได้เสถียรกว่า |
| ขา Analog (ADC) | 1 ขา (10-bit) | 18 ขา (12-bit) | ต่อได้ทั้ง pH, EC และเซ็นเซอร์อื่นๆ พร้อมกัน |
| การเชื่อมต่อ | WiFi เท่านั้น | WiFi + Bluetooth | ใช้ Bluetooth ตั้งค่า WiFi ผ่านมือถือได้ง่าย (Provisioning) |
| ความเร็ว (Clock) | 80 MHz | 160 - 240 MHz | รองรับการประมวลผลคำสั่ง MQTT ที่ซับซ้อนได้รวดเร็ว |

ข้อควรระวังในการใช้งาน (Tips)

แรงดันไฟฟ้า ESP32 ทำงานที่ระดับ 3.3V เท่านั้น หากนำเซ็นเซอร์ 5V มาต่อเข้าขา Input โดยตรง อาจทำให้ชิปพังได้ (ควรใช้ Logic Level Converter)

ขา GPIO ที่จำกัด ขาบางขา (เช่น GPIO 0, 2, 12, 15) มีผลต่อการ Boot เครื่อง (Strapping Pins) ควรหลีกเลี่ยงการต่อ Relay เข้ากับขาเหล่านี้เพื่อป้องกันเครื่องเปิดไม่ติด

2.6.3 โมดูลเซ็นเซอร์วัดอุณหภูมิและความชื้น (DHT22)

DHT22 หรือชื่ออย่างเป็นทางการคือ AM2302 เป็นเซ็นเซอร์วัดค่ามาตรฐานอุณหภูมิและความชื้นสัมพัทธ์แบบดิจิทัล (Wired Digital Output) ที่ได้รับความนิยมในงานด้านเกษตรอัจฉริยะ เนื่องจากมีเสถียรภาพสูงและมีความแม่นยำมากกว่ารุ่น DHT11 อย่างมีนัยสำคัญ

โครงสร้างและหลักการทำงาน (Structure & Working Principle) ภายในตัวถังพลาสติกของ DHT22 ประกอบด้วยส่วนประกอบสำคัญ 3 ส่วน ได้แก่

2.6.3.1 Capacitive Humidity Sensor ใช้สำหรับวัดความชื้นสัมพัทธ์ ทำงานโดยอาศัยหลักการเปลี่ยนแปลงค่าความจุไฟฟ้า (Capacitance) ของสารโพลีเมอร์ที่ดูดซับความชื้น เมื่อความชื้นเปลี่ยน ค่าความจุจะเปลี่ยนตาม

2.6.3.2 NTC Thermistor ใช้สำหรับวัดอุณหภูมิ โดยอาศัยหลักการเปลี่ยนแปลงค่าความต้านทานตามอุณหภูมิ (Negative Temperature Coefficient)

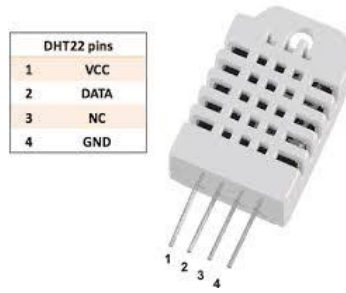
2.6.3.3 8-bit Microcontroller ชิปประมวลผลขนาดเล็กทำหน้าที่แปลงสัญญาณอนาล็อกที่ได้จากตัววัดทั้งสอง ให้กลายเป็นสัญญาณดิจิทัลรูปแบบเฉพาะ (Proprietary Digital Signal) และส่งออกมาทางขา Data

ตารางที่ 2 คุณสมบัติ DHT22

| คุณสมบัติ (Parameter) | รายละเอียด (Specification) |
|---|--|
| แรงดันไฟเลี้ยง (Operating Voltage) | 3.3V - 6V DC (เหมาะสมกับ ESP32 ที่ 3.3V) |
| กระแสไฟฟ้าขณะทำงาน (Operating Current) | 1 - 1.5 mA (ขณะวัดค่า), 40 - 50 μ A (Standby) |
| ช่วงการวัดอุณหภูมิ (Temp Range) | -40°C ถึง +80°C |
| ความแม่นยำอุณหภูมิ (Temp Accuracy) | $\pm 0.5^{\circ}\text{C}$ |
| ความละเอียดอุณหภูมิ (Temp Resolution) | 0.1°C |
| ช่วงการวัดความชื้น (Humidity Range) | 0% ถึง 100% RH |
| ความแม่นยำความชื้น (Humidity Accuracy) | $\pm 2\% \text{ RH}$ (สูงสุด $\pm 5\% \text{ RH}$) |
| ความละเอียดความชื้น (Humidity Resolution) | $0.1\% \text{ RH}$ |
| ระยะเวลาในการสุ่มวัด (Sampling Period) | 2 วินาที (0.5 Hz) |
| โปรโตคอลสื่อสาร (Communication) | Single-Bus (1-Wire protocol แต่ไม่ใช่ Dallas 1-Wire) |

รายละเอียดขาใช้งาน (Pin Configuration)

โมดูล DHT22 แบบที่มีบอร์ด PCB รองรับ (Module Type) มักจะมี 3 ขา แต่หากเป็นตัวเซ็นเซอร์เปล่าจะมี 4 ขา ดังนี้:



ภาพที่ 4 โมดูลเซ็นเซอร์วัดอุณหภูมิและความชื้น

Pin 1 (VCC): ต่อไฟเลี้ยง 3.3V - 5V

Pin 2 (DATA): ขาส่งสัญญาณข้อมูลดิจิทัล (Serial Data) จำเป็นต้องมีตัวต้านทาน Pull-up ($4.7k\Omega - 10k\Omega$) เชื่อมไปยัง VCC

Pin 3 (NC): ไม่มีการเชื่อมต่อใช้งาน (Null)

Pin 4 (GND): ต่อกราวด์

2.6.3.4 โปรโตคอลการรับส่งข้อมูล (Data Transmission Protocol)

การสื่อสารระหว่าง MCU (ESP32) และ DHT22 ใช้สายสัญญาณเพียงเส้นเดียว โดยมีการส่งข้อมูลขนาด 40 bit ต่อครั้ง ประกอบด้วย:

1. Start Signal MCU ดึงแรงดันลงกราวด์ (Low) อย่างน้อย 18ms เพื่อปลุกเซ็นเซอร์
2. Response DHT22 ส่งสัญญาณตอบรับ
3. Data Transmission (40 Bits):
 - 16 bits แรก ค่าความชื้น (Humidity Data)
 - 16 bits ถัดมา ค่าอุณหภูมิ (Temperature Data)
 - 8 bits สุดท้าย ค่าตรวจสอบความถูกต้อง (Parity Checksum)

สูตรตรวจสอบความถูกต้อง (Checksum)

$$\text{Checksum} = (\text{High_Hum} + \text{Low_Hum} + \text{High_Temp} + \text{Low_Temp}) \ \& \ 0xFF$$

ความสำคัญต่องานวิจัย (Application in Research)

ในโรงงาน Plant Factory สำหรับสตรอเบอร์รี่ DHT22 มีความสำคัญอย่างยิ่งในด้าน:

2.6.3.5 การคุมจุดน้ำค้าง (Dew Point Control): เพื่อป้องกันเชื้อราและความชื้นสัมพัทธ์ที่สูงเกินไปจนพืชคายน้ำไม่ได้ ซึ่งเป็นสาเหตุหลักของอาการขอบใบไหม้ (Tip Burn)

2.6.3.6 ความแม่นยำระดับทศนิยม: ความละเอียด 0.1 ช่วยให้ระบบ PID Control หรือ Threshold Control ทำงานได้ละเอียดอ่อนกว่าการใช้ DHT11

2.6.4 หัววัดค่าความเป็นกรด-ด่าง (pH Probe)

pH Probe คือเซ็นเซอร์เคมีไฟฟ้า (Electrochemical Sensor) ที่ทำหน้าที่ตรวจวัดค่าความเข้มข้นของไฮโดรเจนไอออน (H^+) ในสารละลาย ซึ่งเป็นปัจจัยวิกฤตในการปลูกสตรอเบอร์รี่ (ค่าที่เหมาะสมคือ pH 5.5 - 6.5) เพื่อให้รากพืชสามารถดูดซึมธาตุอาหารได้สูงสุด



ภาพที่ 5 เซ็นเซอร์เคมีไฟฟ้า pH Probe

โครงสร้างและส่วนประกอบ (Structure & Components)

หัววัด pH ทั่วไปที่ใช้ในงานเกษตรเป็นแบบ Combination Electrode คือรวมขั้วไฟฟ้า 2 ชนิดไว้ในแท่งแก้วเดียวกัน ประกอบด้วย:

2.6.4.1 Sensing Electrode (ขั้ววัด) ส่วนปลายทำจากแก้วชนิดพิเศษ (pH-sensitive glass membrane) ภายในบรรจุสารละลายบัฟเฟอร์ pH 7 และเส้นลวดเงินเคลือบซิลเวอร์คลอไรด์ (Ag/AgCl)

2.6.4.2 Reference Electrode (ขั้วอ้างอิง) ทำหน้าที่สร้างศักย์ไฟฟ้าคงที่อ้างอิง โดยทั่วไปจะใช้ Ag/AgCl แช่อยู่ในสารละลายอิเล็กโทรไลต์ (KCl 3 mol/L)

2.6.4.3 Junction (จุดเชื่อมต่อ) มักเป็นเซรามิกรูพรุน ยอมให้ไอออนเคลื่อนที่ผ่านได้เพื่อให้วงจรไฟฟ้าสมบูรณ์

2.6.4.4 Body ตัวถังทำจากพลาสติก (Epoxy/PE) หรือแก้ว เพื่อปกป้องชิ้นส่วนภายใน

หลักการทำงาน (Working Principle)

pH Probe ทำงานโดยอาศัยหลักการวัด ความต่างศักย์ไฟฟ้า (Potential Difference) ที่เกิดขึ้นระหว่าง Sensing Electrode และ Reference Electrode ซึ่งเป็นไปตาม สมการเนิร์นสต์ (Nernst Equation):

$$E = E^0 - \frac{2.303RT}{nF} (pH_{unknown} - pH_{internal})$$

ภาพที่ 6 สูตรคำนวณค่า pH Probe

E = แรงดันไฟฟ้าที่วัดได้ (Voltage)

T = อุณหภูมิสัมบูรณ์ (Kelvin)

R, F = ค่าคงที่แก๊สและค่าคงที่ฟาราเดย์

n = จำนวนประจุ (สำหรับ H⁺ เท่ากับ 1)

ในทางปฏิบัติ: ที่อุณหภูมิ 25°C ค่าความต่างศักย์จะเปลี่ยนแปลงประมาณ -59.16 mV ต่อ 1 หน่วย pH

pH 7 (เป็นกลาง): แรงดันไฟฟ้า 0 mV

pH < 7 (กรด): แรงดันไฟฟ้าเป็น บวก (เช่น +177 mV ที่ pH 4)

pH > 7 (ด่าง): แรงดันไฟฟ้าเป็น ลบ (เช่น -177 mV ที่ pH 10)

ข้อมูลจำเพาะทางเทคนิค (Technical Specifications)

เนื่องจากสัญญาณที่ออกจากหัววัด (Probe) มีค่าความต้านทานสูงมาก (High Impedance) จึงต้องใช้งานร่วมกับ Signal Interface Board (Op-Amp Module) เพื่อขยายสัญญาณให้ ESP32 อ่านค่าได้ รายละเอียดทั่วไปมีดังนี้

| คุณสมบัติ (Parameter) | รายละเอียด (Specification) |
|--|--------------------------------------|
| ช่วงการวัด (Measuring Range) | 0.00 - 14.00 pH |
| จุดศูนย์ (Zero Point) | ± 0.5 pH (ที่ 0 mV) |
| ความชันทางทฤษฎี (Theoretical Slope) | ≈ -59.16 mV/pH (ที่ 25°C) |
| ความแม่นยำ (Accuracy) | ± 0.1 pH (ขึ้นอยู่กับ การ Calibrate) |
| ความต้านทานภายใน (Internal Resistance) | ≤ 250 MΩ (ที่ 25°C) |
| เวลาตอบสนอง (Response Time) | ≤ 1 นาที (วัดถึง 95% ของค่าจริง) |
| อุณหภูมิใช้งาน (Operating Temp) | 0 - 60°C |
| ชนิดหัวต่อ (Connector Type) | BNC (Bayonet Neill-Concelman) |

ตารางที่ 3 คุณสมบัติ Parameter

2.6.4.5 การเชื่อมต่อสัญญาณ (Signal Interface Module) หัววัด pH ไม่สามารถต่อเข้ากับ ESP32 โดยตรงได้ เนื่องจาก แรงดัน Output มีค่าติดลบ (เช่น -177mV) แต่ ESP32 รับได้แค่ $0-3.3\text{V}$ Impedance สูงมาก ทำให้กระแสต่ำจน ADC อ่านไม่ได้ ดังนั้นจึงต้องใช้ Module แปลงสัญญาณ (pH Circuit Board) ซึ่งมักใช้ Op-Amp เบอร์เช่น

2.6.4.6 TL082 หรือ LM358 ทำหน้าที่ Amplification ขยายสัญญาณ mV ให้อยู่ในช่วง Volt

Offset Adjustment ยกระดับแรงดันจากลบให้กลายเป็นบวก (เช่น ตั้งให้ $\text{pH } 7 = 1.65\text{V}$ หรือ 2.5V กลางของช่วงไฟเลี้ยง)

2.6.4.7 การบำรุงรักษาและข้อควรระวัง (Maintenance & Caution) ในงานวิจัยระยะยาว เรื่องนี้สำคัญมากและควรระบุในบทที่ 2 หรือ 3

2.6.4.8 การเก็บรักษา หัววัดต้องแช่อยู่ในน้ำยา $\text{KCl } 3\text{M}$ (Potassium Chloride) เสมอ ห้าม แช่ในน้ำกลั่น (Distilled Water) เพราะจะทำให้อิเล็กโทรไลต์ภายในเจือจางและหัววัดเสียสภาพ

2.6.4.9 การทำความสะอาด: ใช้น้ำเปล่าล้างสารเคมีตกค้างและซับเบาๆ ด้วยทิชชู ห้ามถูแรงๆ เพราะจะเกิดไฟฟ้าสถิตรบกวนการอ่านค่า Calibration จำเป็นต้องสอบเทียบ (Calibrate) อย่างน้อยทุก 1

2 สัปดาห์ ด้วยสารละลายบัฟเฟอร์มาตรฐาน (Buffer Solution) ค่า 4.01 และ 6.86 เพื่อปรับค่า Slope และ Offset ในโปรแกรม

2.6.5 เซ็นเซอร์วัดค่าการนำไฟฟ้า (TDS Sensor)



ภาพที่ 7 เซ็นเซอร์วัดค่าการนำไฟฟ้า TDS Sensor

ในการปลูกพืชระบบปิด การควบคุมธาตุอาหารให้เหมาะสมเป็นปัจจัยหลักในการกำหนดคุณภาพผลผลิต โดยเครื่องมือที่ใช้ตรวจวัดความเข้มข้นของสารละลายธาตุอาหารคือ EC Sensor (Electrical Conductivity) หรือ TDS Sensor (Total Dissolved Solids) ซึ่งทั้งสองค่านี้มีความสัมพันธ์กันทางคณิตศาสตร์นิยามและความแตกต่าง (Definition & Difference)

2.6.5.1 Electrical Conductivity (EC) คือความสามารถของน้ำในการนำกระแสไฟฟ้า ยิ่งมีแร่ธาตุ (ไอออนบวกและลบ) เจือปนมาก น้ำจะยิ่งนำไฟฟ้าได้ดี หน่วยวัดคือ mS/cm (มิลลิซีเมนส์ต่อเซนติเมตร) หรือ $\mu\text{S}/\text{cm}$ (ไมโครซีเมนส์ต่อเซนติเมตร)

2.6.5.2 Total Dissolved Solids (TDS) คือค่าประมาณของน้ำหนักรวมของของแข็งที่ละลายอยู่ในน้ำ โดยคำนวณแปลงมาจากค่า EC หน่วยวัดคือ ppm (Part Per Million หรือ มิลลิกรัมต่อลิตร)

2.6.5.3 สมการแปลงค่าโดยประมาณ $\text{TDS (ppm)} \approx \text{EC } (\mu\text{S}/\text{cm}) \times 0.5$
(ค่าตัวคูณ 0.5 - 0.7 ขึ้นอยู่กับประเภทของเกลือ)

โครงสร้างและหลักการทำงาน (Structure & Working Principle)
เซ็นเซอร์ทำงานโดยอาศัยหลักการ Conductometry

2.6.5.4 Electrodes (ขั้วไฟฟ้า) หัววัดประกอบด้วยขั้วไฟฟ้า 2 ขั้ว (หรือ 4 ขั้วในรุ่นอุตสาหกรรม) วางห่างกันในระยะที่แน่นอน

2.6.5.5 Excitation วงจรจะปล่อยแรงดันไฟฟ้ากระแสสลับ (AC) หรือกระแสตรงแบบพัลส์ (Pulsed DC) ผ่านขั้วไฟฟ้าหนึ่งไปยังอีกขั้วหนึ่งผ่านน้ำ

หมายเหตุ การใช้กระแสสลับ (AC) ช่วยลดปัญหา Polarization หรือการเกาะตัวของประจุที่ขั้ววัด ซึ่งจะทำให้ค่าเพี้ยนและขั้วสึกกร่อน

2.6.5.6 Measurement วัดกระแสไฟฟ้าที่ไหลผ่าน หากมีไอออน (ปุ๋ย) มาก ความต้านทานจะต่ำ กระแสจะไหลได้มาก

2.6.5.7 Signal Processing วงจรขยายสัญญาณ (Op-Amp) จะแปลงค่ากระแสที่วัดได้เป็นแรงดันไฟฟ้า (Analog Voltage 0-2.3V หรือ 0-5V) ส่งไปยัง ESP32

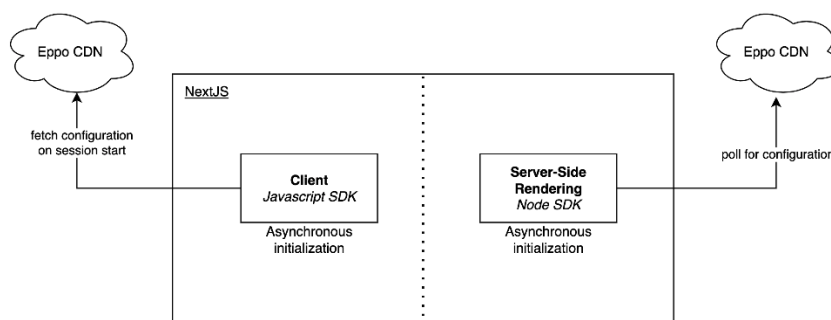
2.6.6 เฟรมเวิร์กสำหรับการพัฒนาเว็บแอปพลิเคชัน (Next.js)

NEXT.js

ภาพที่ 8 เฟรมเวิร์กสำหรับการพัฒนาเว็บแอปพลิเคชัน

Next.js คือ Open-source Web Development Framework ที่สร้างขึ้นบนพื้นฐานของ React (JavaScript Library) พัฒนาโดยบริษัท Vercel ถูกออกแบบมาเพื่อแก้ปัญหาเรื่องประสิทธิภาพ (Performance) และการจัดการ SEO (Search Engine Optimization) ที่มักพบใน Single Page Application (SPA) ทั่วไป โดยในโครงงานนี้ Next.js ทำหน้าที่เป็น User Interface (UI) สำหรับ Dashboard ควบคุมโรงเรือน

2.6.6.1 สถาปัตยกรรมหลัก (Core Architecture) Next.js มีจุดเด่นทางเทคนิคที่ผสมผสานการทำงานระหว่างฝั่งเซิร์ฟเวอร์ (Server) และฝั่งผู้ใช้งาน (Client) ดังนี้:



ภาพที่ 9 การทำงานของสถาปัตยกรรมหลัก

2.6.6.2 Server-Side Rendering (SSR) การประมวลผลหน้าเว็บให้เสร็จสมบูรณ์จากฝั่งเซิร์ฟเวอร์ ก่อนส่งมายังเบราว์เซอร์ ทำให้หน้า Dashboard โหลดขึ้นมาแสดงผลได้ทันที (First Contentful Paint เร็วมาก) ผู้ใช้งานไม่ต้องรอรอก่อนหน้าจะโหลด

2.6.6.3 Client-Side Rendering (CSR) หลังจากโหลดหน้าเว็บครั้งแรกเสร็จสิ้น Next.js จะเปลี่ยนโหมดการทำงานเป็น CSR (ผ่านกระบวนการที่เรียกว่า Hydration) เพื่อให้การโต้ตอบ เช่น การกดปุ่มเปิด-ปิด บั๊มน้ำ หรือการดูกราฟ Real-time เป็นไปอย่างลื่นไหลโดยไม่ต้องรีโหลดหน้าใหม่

2.6.6.4 Component-Based Architecture การพัฒนาแบ่งออกเป็นส่วนย่อยๆ (Components) เช่น ส่วนแสดงอุณหภูมิ, ส่วนกราฟ, ส่วนปุ่มควบคุม ทำให้โค้ดเป็นระเบียบ แก้ไขง่าย และสามารถนำกลับมาใช้ซ้ำ (Reusable) ได้

2.6.6.5 ฟีเจอร์ทางเทคนิคที่สำคัญ (Key Technical Features) A. File-system Based Routing (App Router) Next.js ใช้ระบบจัดการเส้นทาง (Routing) โดยอิงตามโครงสร้างโฟลเดอร์และไฟล์ ไม่ต้องเขียน Config Router ให้ยุ่งยาก

2.6.6.5.1 ตัวอย่าง การสร้างไฟล์ app/dashboard/page.js จะกลายเป็น URL www.yoursite.com/dashboard โดยอัตโนมัติ ช่วยให้โครงสร้างโปรเจกต์เข้าใจง่าย

B. API Routes (Backend-for-Frontend) Next.js สามารถทำหน้าที่เป็น Backend ขนาดย่อมได้ผ่าน API Routes การใช้งานในโครงการ: สามารถสร้าง Endpoint (เช่น /api/control) เพื่อรับคำสั่งจากหน้าเว็บ แล้วส่งต่อให้ MQTT Broker หรือ Firebase เพื่อส่งงานฮาร์ดแวร์ ESP32 อีกทอดหนึ่ง เป็นการเพิ่มความปลอดภัยโดยไม่ให้ Frontend ต่อกับ Database โดยตรง C. Data Fetching & Caching รองรับการดึงข้อมูลที่ยืดหยุ่น ทั้งแบบ Static (ดึงตอน Build) และ Dynamic (ดึงตอน Request)

2.6.6.6 การใช้งาน ใช้ร่วมกับไลบรารีอย่าง SWR หรือ TanStack Query เพื่อดึงข้อมูลสถานะเซ็นเซอร์ (Temp, Humid, pH) จาก Firebase มาแสดงผลแบบ Real-time โดยไม่กินทรัพยากรเครื่องมากเกินไป

2.6.6.7 การประยุกต์ใช้ในระบบ Smart Farm Dashboard ในโครงการนี้ Next.js ถูกใช้เพื่อสร้าง Web Dashboard ที่มีฟังก์ชันดังนี้

2.6.6.7.1 Real-time Monitoring แสดงค่าจากเซ็นเซอร์ (DHT22, pH, EC) เป็นกราฟเส้นหรือเกจวัด (Gauge Chart) โดยใช้ไลบรารีเสริม เช่น Recharts หรือ Chart.js

2.6.6.7.2 Control Interface สร้างปุ่มสวิตช์ (Toggle Switch) สำหรับสั่งเปิด-ปิด ไฟ LED, พัดลม และปั๊มน้ำ ผ่านโปรโตคอล MQTT

2.6.6.7.3 Alert System แสดง Pop-up หรือ Notification แจ้งเตือนเมื่อค่าวิกฤต (เช่น pH ต่ำเกินไป หรือตรวจพบโรคพืชจาก AI)

2.6.6.7.4 Responsive Design: รองรับการใช้งานทั้งบนคอมพิวเตอร์และสมาร์ทโฟน ช่วยให้เกษตรกรดูแลฟาร์มได้จากทุกที่

ตารางเปรียบเทียบกับเทคโนโลยีอื่น (Comparison)

| คุณสมบัติ | React (Standard) | Next.js (Framework) | ประโยชน์ในโครงการ | คุณสมบัติ |
|-------------|---------------------------|-----------------------|---|-------------|
| การเรนเดอร์ | Client-Side (CSR) | Hybrid (SSR + CSR) | Dashboard โหลดเร็ว ขึ้น ไม่ขาวโพลน ตอนเน็ตช้า | การเรนเดอร์ |
| Routing | ต้องลง Library เพิ่ม | มีในตัว (File-system) | พัฒนาได้รวดเร็ว จัดการหน้าง่าย | Routing |
| API Backend | ทำไม่ได้ (ต้องแยก Server) | มี API Routes ในตัว | ซ่อน Logic การเชื่อมต่อ MQTT/Database ได้ | API Backend |

ตารางที่ 4 ความแตกต่าง React และ Next.js

2.6.7.1 ภาษาไพธอน (Python Programming Language)



Python คือภาษาโปรแกรมระดับสูง (High-level Programming Language) ชนิด Interpreted Language ที่ถูกออกแบบมาให้อ่านเข้าใจง่าย มีโครงสร้างที่ไม่ซับซ้อน แต่ทรงประสิทธิภาพสูงในการประมวลผลข้อมูล ปัจจุบัน Python ได้รับการยอมรับว่าเป็นภาษามาตรฐานอันดับหนึ่งสำหรับงานด้านปัญญาประดิษฐ์ (Artificial Intelligence) และวิทยาศาสตร์ข้อมูล (Data Science)

2.6.7.1 คุณลักษณะทางเทคนิค (Technical Characteristics) Interpreted & Dynamic Typing Python ทำงานโดยใช้ตัวแปลภาษา (Interpreter) ที่ละบรรทัด ทำให้ไม่ต้องผ่านกระบวนการ Compile ที่ยุ่งยากเหมือน C++ ช่วยให้การพัฒนาและทดสอบ (Debug) ระบบ AI ทำได้รวดเร็ว

2.6.7.2 Extensive Libraries จุดเด่นที่สุดคือ Ecosystem ที่มีไลบรารีพร้อมใช้จำนวนมากสำหรับงานเฉพาะทาง เช่น การจัดการเมทริกซ์ภาพ (NumPy), การประมวลผลภาพ (OpenCV), และ Deep Learning (PyTorch/TensorFlow)

2.6.7.3 Cross-Platform สามารถรันได้บนหลายระบบปฏิบัติการ (Windows, Linux, macOS) ทำให้สามารถย้ายโค้ดจากเครื่องคอมพิวเตอร์ที่ใช้ Train โมเดล ไปรันบนอุปกรณ์ขนาดเล็กอย่าง Raspberry Pi หรือ Jetson Nano ได้ในอนาคต

2.6.7.3 บทบาทในโครงการ (Role in Project) ในระบบ Plant Factory อัจฉริยะนี้ Python ไม่ได้ทำหน้าที่เป็นเพียงสคริปต์ธรรมดา แต่ทำหน้าที่เป็น "AI Processing Node" โดยมีกระบวนการทำงานดังนี้

2.6.7.3.1 Image Acquisition รับข้อมูลภาพแบบ Stream หรือ Snapshot จาก ESP32-CAM ผ่านโปรโตคอล HTTP

2.6.7.3.2 Preprocessing ใช้ไลบรารี OpenCV ในการปรับขนาดภาพ (Resize), แปลงระบบสี (Color Space Conversion), และลดสัญญาณรบกวนก่อนส่งเข้าโมเดล

2.6.7.3.3 Inference Engine โหลดโมเดล YOLOv8 ที่ผ่านการเรียนรู้ (Training) แล้ว เพื่อทำการตรวจจับวัตถุ (Object Detection) โดยจำแนกความสุกของสตรอเบอร์รี่ และอาการผิดปกติ

2.6.7.3.4 Decision Making นำผลลัพธ์จาก AI มาเข้าสู่เงื่อนไขทางตรรกะ (Logic) เช่น หากพบผลสุก > 70% จำนวน 3 ผลขึ้นไป ให้ส่งคำสั่งปรับค่า EC

2.6.7.3.5 Connectivity ส่งคำสั่งควบคุมกลับไปยัง ESP32 ผ่านโปรโตคอล MQTT

2.6.7.4 ไลบรารีสำคัญที่เลือกใช้ (Key Libraries Utilized) เพื่อให้ระบบทำงานได้สมบูรณ์ โครงการนี้มีการเรียกใช้ไลบรารีเฉพาะทางดังนี้

2.6.7.4.1 Ultralytics (YOLOv8) หน้าที่เป็น Framework หลักในการรันโมเดล Object Detection จุดเด่นทางเทคนิค YOLOv8 (You Only Look Once version 8) เป็นโมเดลประเภท One-stage detector ที่มีความเร็วในการประมวลผลสูงมาก (High FPS) เหมาะสำหรับงาน Real-time และมีความแม่นยำ (mAP) สูงกว่ารุ่นก่อนหน้าอย่าง YOLOv5

2.6.7.4.2 OpenCV (Open Source Computer Vision Library) หน้าที่: จัดการข้อมูลภาพดิจิทัล การใช้งาน ใช้ฟังก์ชัน `cv2.VideoCapture` เพื่อดึงภาพจากกล้อง, `cv2.resize` เพื่อปรับขนาดภาพให้เข้ากับ Input size ของโมเดล (เช่น 640x640 pixel), และวาดกรอบสี่เหลี่ยม (Bounding Box) แสดงผลลัพธ์บนหน้าจอ

2.6.7.4.3 Paho-MQTT หน้าที่: จัดการการสื่อสารข้อมูลระหว่าง Python และอุปกรณ์ IoT การทำงาน ทำหน้าที่เป็น MQTT Client ที่เชื่อมต่อกับ Broker เพื่อ Publish คำสั่ง (Command) และ Subscribe ค่าสถานะ (Status)

2.6.7.4.4 NumPy หน้าที่: การคำนวณทางคณิตศาสตร์และจัดการอาร์เรย์ (Array) ความสำคัญ ภาพดิจิทัลในคอมพิวเตอร์คือ "เมทริกซ์ของตัวเลข" (Pixel Values) NumPy จึงเป็นตัวกลางที่ช่วยแปลงข้อมูลภาพให้ AI สามารถนำไปคำนวณต่อได้



ภาพที่ 11 โปรแกรมแก้ไขซอร์สโค้ด VS Code

2.6.8 โปรแกรมวิชวลสตูดิโอโค้ด (Visual Studio Code)

Visual Studio Code (หรือ VS Code) คือโปรแกรมแก้ไขซอร์สโค้ด (Source Code Editor) ที่พัฒนาโดยบริษัท Microsoft ซึ่งทำงานภายใต้สัญญาอนุญาตแบบ Open Source (MIT License) โดยมีความแตกต่างจาก IDE (Integrated Development Environment) แบบดั้งเดิมตรงที่มีความเบา (Lightweight) ยืดหยุ่นสูง และรองรับการทำงานข้ามแพลตฟอร์ม (Cross-Platform) ทั้ง Windows, macOS และ Linux

2.6.8.1 สถาปัตยกรรมระบบ (System Architecture) VS Code ไม่ได้ถูกเขียนขึ้นด้วยภาษา Native แบบโปรแกรมทั่วไป แต่ถูกสร้างขึ้นบนเฟรมเวิร์ก Electron ซึ่งเป็นการผสมผสานเทคโนโลยีเว็บเข้ากับแอปพลิเคชันเดสก์ท็อป โดยมีองค์ประกอบหลักทางเทคนิคดังนี้:

2.6.8.2 Electron Framework ช่วยให้ VS Code ทำงานได้บน Chromium (Rendering Engine) และ Node.js (Runtime Environment) ทำให้สามารถใช้ JavaScript/TypeScript ในการควบคุมการเข้าถึงไฟล์ระบบ (File System) และเครือข่ายได้

2.6.8.3 Monaco Editor เป็น Core Editor component ที่อยู่เบื้องหลัง VS Code ทำหน้าที่จัดการการแสดงผลข้อความ ไฮไลต์สี (Syntax Highlighting) และจัดการคีย์ลัด

2.6.8.4 Extension Host Process VS Code แยกกระบวนการทำงานของ "ส่วนขยาย" (Extensions) ออกจาก ตัวโปรแกรมหลัก (Core) อย่างชัดเจน ข้อดีทางเทคนิคคือ หากส่วนขยายใดทำงานผิดพลาดหรือค้าง (Crash) จะไม่ส่งผลกระทบต่อตัวโปรแกรมหลัก ทำให้ Editor ยังคงทำงานต่อได้โดยไม่ปิดตัวเอง

2.6.8.5 เทคโนโลยีโปรโตคอลภาษา (Language Server Protocol - LSP) หนึ่งในนวัตกรรมสำคัญที่ควรระบุในงานวิจัยคือการใช้ LSP ซึ่งทำให้ VS Code รองรับภาษาได้เกือบทุกภาษาในโลก โดยไม่ต้องติดตั้งตัวแปลภาษาทั้งหมดลงในโปรแกรมหลัก

2.6.8.6 หลักการทำงาน แทนที่ VS Code จะต้องรู้วิธีการเขียนโค้ด C++ หรือ Python ด้วยตัวเอง มันจะทำตัวเป็น "Client" ที่ส่งข้อมูลโค้ดไปยัง "Language Server" (ซึ่งเป็นโปรแกรมแยกต่างหาก)

2.6.8.7 ผลลัพธ์ Language Server จะประมวลผลและส่งข้อมูลกลับมาเพื่อทำฟีเจอร์อัจฉริยะ เช่น IntelliSense (การเติมคำอัตโนมัติ), Go to Definition (กระโดดไปหาต้นตอของฟังก์ชัน), และ Error Checking (แจ้งเตือนจุดผิดพลาดทันทีขณะพิมพ์)

2.6.8.8 ส่วนขยายที่ใช้ในงานวิจัย (Essential Extensions Integration) เนื่องจากโครงการนี้เป็นระบบ IoT ที่ซับซ้อน VS Code จึงถูกปรับแต่งผ่าน Extensions เพื่อให้รองรับงานเฉพาะด้าน

2.6.8.9 PlatformIO IDE (สำหรับ Embedded System) รายละเอียด เป็นระบบ Ecosystem สำหรับการพัฒนา IoT ที่เข้ามาแทนที่ Arduino IDE เดิม เทคนิค PlatformIO มีระบบจัดการ Library อัตโนมัติ (Library Dependency Management) และใช้ Build System ที่ตรวจสอบความเข้ากันได้ของฮาร์ดแวร์ ทำให้สามารถคอมไพล์โค้ดสำหรับ ESP32 ได้รวดเร็วและจัดการพาร์ติชันหน่วยความจำได้ดีกว่า

2.6.8.10 Python & Pylance (สำหรับ AI Backend) รายละเอียด ส่วนขยายที่ช่วยในการเขียน Python สำหรับระบบ AI เทคนิค ใช้เทคโนโลยี Analysis Engine เพื่อวิเคราะห์ Type Checking ของตัวแปรในโมเดล YOLOv8 และ OpenCV ช่วยลดข้อผิดพลาดขณะรันไทม์ (Runtime Errors)

2.6.8.11 Git Graph (สำหรับ Version Control) รายละเอียด เครื่องมือสำหรับจัดการประวัติการแก้ไขโค้ด เทคนิค แสดงผล Branching Model ของ Git ในรูปแบบกราฟิก ทำให้ง่ายต่อการย้อนกลับเวอร์ชัน (Revert) หากโค้ดใหม่มีปัญหา

2.6.8.12 โปรโตคอลการดีบั๊ก (Debug Adapter Protocol - DAP) นอกจาก LSP แล้ว VS Code ยังใช้ DAP ซึ่งเป็นโปรโตคอลมาตรฐานที่ช่วยให้ผู้พัฒนาสามารถ Debug (ตรวจสอบจุดผิดพลาดที่ละบรรทัด) ได้หลายภาษาในหน้าจอเดียว ประโยชน์ในโครงการ สามารถตั้ง Breakpoint เพื่อดูค่าตัวแปรในโค้ด Python ขณะที่กำลังรับภาพจากกล้อง หรือตรวจสอบค่า Memory ใน ESP32 (เมื่อต่อผ่าน JTAG) ได้โดยไม่ต้องเปลี่ยนโปรแกรม

2.7 รายละเอียดของโครงการ

โครงการนี้เป็นการบูรณาการระหว่างวิศวกรรมคอมพิวเตอร์และเกษตรอัจฉริยะ โดยหัวใจหลักคือระบบ Soft Sensors ซึ่งใช้ภาพจากกล้องแทนเซ็นเซอร์กายภาพบางส่วน ระบบจะวิเคราะห์ภาพถ่ายผลสตรอเบอร์รี่ผ่านโมเดล YOLOv8 หากพบว่ามีอัตราส่วนผลสุกเกิน 70% ระบบ Python จะส่งสัญญาณสั่งการผ่าน MQTT ไปยัง ESP32 ให้ปรับเพิ่มระดับโพแทสเซียมในสารละลายทันที เพื่อเร่งการสะสมน้ำตาลในช่วงสุดท้ายก่อนการเก็บเกี่ยว (Pre-harvest optimization) นอกจากนี้ยังมีการจัดการความร้อนเชิงวิศวกรรมโดยแยกชุดหม้อแปลงไว้นอกกล่องปลุกและควบคุมการหมุนเวียนอากาศ (Air change) ให้สม่ำเสมอเพื่อป้องกันความชื้นสะสมที่ปลายใบ

บทที่ 3

สรุปผลการปฏิบัติงาน

3.1 สิ่งที่คาดหวัง

ในการเข้าปฏิบัติงานสหกิจศึกษาครั้งนี้ สิ่งที่คาดหวังคือการได้นำความรู้ด้านวิศวกรรมคอมพิวเตอร์มาประยุกต์ใช้ในสถานการณ์จริง โดยเฉพาะการบูรณาการระบบควบคุม (IoT) เข้ากับปัญญาประดิษฐ์ (AI) เพื่อแก้ปัญหาทางด้านเกษตรกรรมแม่นยำ นอกจากนี้ยังคาดหวังที่จะเข้าใจกระบวนการวิจัยและพัฒนาผลิตภัณฑ์ ตั้งแต่การออกแบบโครงสร้างเชิงวิศวกรรมไปจนถึงการพัฒนาส่วนติดต่อผู้ใช้งานที่ตอบโจทย์กลุ่มเป้าหมาย

3.2 ประโยชน์ที่ได้รับจากการปฏิบัติงาน

3.2.1 ประโยชน์ต่อตนเอง

3.2.1.1 ได้ทักษะการพัฒนา Full-stack Web Application ด้วย Next.js และการจัดการฐานข้อมูลแบบ Real-time

3.2.1.2 มีความเชี่ยวชาญในการฝึกสอน (Training) และปรับจูนโมเดล YOLOv8 เพื่อใช้งานในสภาพแวดล้อมจริง

3.2.1.3 เข้าใจหลักการทำงานของเซ็นเซอร์ด้านการเกษตร (pH, EC, PPFD) และเทคนิคการเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ ESP32 ระดับสูง

3.2.2 ประโยชน์ต่อสถานประกอบการ

3.2.2.1 สถานประกอบการได้รับต้นแบบระบบปลูกสตรอเบอรี่อัจฉริยะที่สามารถทำงานได้โดยอัตโนมัติ ช่วยลดความเสี่ยงจากการใช้ประสบการณ์มนุษย์เพียงอย่างเดียว

3.2.2.2 ได้รับระบบฐานข้อมูลพฤติกรรมการเติบโตของสตรอเบอรี่พันธุ์พระราชทาน 80 ซึ่งสามารถนำไปต่อยอดงานวิจัยเชิงพาณิชย์ได้

3.2.2.3 ระบบ Feedback Loop ที่พัฒนาขึ้นช่วยเพิ่มคุณภาพผลผลิต (ค่า TSS) ให้มีความสม่ำเสมอมากขึ้น

3.2.3 ประโยชน์ต่อมหาวิทยาลัย

3.2.3.1 เป็นการสร้างความร่วมมือทางวิชาการและวิจัยระหว่างมหาวิทยาลัยกาฬสินธุ์และมหาวิทยาลัยอุบลราชธานี

3.2.3.2 เป็นแนวทางและกรณีศึกษาให้นักศึกษารุ่นหลังในการประยุกต์ใช้ปัญญาประดิษฐ์ในงานด้านวิศวกรรมสหสาขาวิชา (Multidisciplinary Engineering)

3.3 วิเคราะห์จุดเด่น จุดด้อย โอกาส อุปสรรค (Swot Analysis)

จุดเด่น (Strengths) มีทักษะพื้นฐานด้านการเขียนโปรแกรม Python และ C++ ที่แข็งแกร่ง ทำให้สามารถเชื่อมต่อระบบ AI และ IoT ได้อย่างรวดเร็ว รวมถึงมีความกระตือรือร้นในการเรียนรู้เทคโนโลยีใหม่ เช่น Next.js

จุดด้อย (Weaknesses) ขาดประสบการณ์ด้านพฤกษศาสตร์ในช่วงเริ่มต้น ทำให้ต้องใช้เวลาศึกษาทำความเข้าใจเรื่องความต้องการสารอาหารของพืชเพิ่มเติมเพื่อให้การออกแบบระบบมีความแม่นยำ

โอกาส (Opportunities) การได้รับคำปรึกษาจากผู้เชี่ยวชาญในภาควิชาวิศวกรรมอุตสาหการ และการเข้าถึงอุปกรณ์เครื่องมือวัดระดับอุตสาหกรรม ทำให้ผลงานมีความเป็นมืออาชีพสูงขึ้น

อุปสรรค (Threats) ข้อจำกัดด้านเวลาในการเก็บข้อมูลผลผลิตสตรอเบอรี่ซึ่งต้องใช้เวลาตามรอบการเจริญเติบโตของพืชจริง และปัญหาความแปรปรวนของสัญญาณอินเทอร์เน็ตในบางช่วงของการทดสอบระบบ Cloud

3.3.1 ประสบการณ์ที่ประทับใจ / ประสบการณ์พิเศษ

ประสบการณ์ที่ประทับใจที่สุดคือการที่ระบบ Feedback Loop สามารถทำงานได้อย่างสมบูรณ์เมื่อ AI ตรวจจับผลสุกแล้วสั่งการเพิ่มค่า EC สารอาหารได้สำเร็จตามตรรกะที่วางไว้ ซึ่งเป็นการพิสูจน์ให้เห็นว่าปัญญาประดิษฐ์สามารถช่วยเพิ่มมูลค่าให้กับผลผลิตทางการเกษตรได้อย่างเป็นรูปธรรม นอกจากนี้ยังได้รับมิตรภาพและการถ่ายทอดความรู้ที่มีค่าจากพี่เลี้ยงและเพื่อนร่วมงานในต่างสถาบัน

บทที่ 4
ปัญหาและข้อเสนอแนะ

สถานประกอบการ

ปัญหา

1.....

2.....

ข้อเสนอแนะ

1.....

2.....

มหาวิทยาลัย

ปัญหา

1.....

2.....

ข้อเสนอแนะ

1.....

2.....

นักศึกษา

ปัญหา

1.....

2.....

ข้อเสนอแนะ

1.....

2.....

บรรณานุกรม

ภาคผนวก ก

ภาพถ่ายสถานที่ปฏิบัติงานสหกิจศึกษา ณ บริษัท.....
(ตามที่บริษัทอนุญาตให้เผยแพร่ได้)