



โครงการ สาขา วิทยาศาสตร์สิ่งแวดล้อม
ระบบเลี้ยงกุ้งฝอย IoT ด้วยบอร์ด ESP32

โดย

เด็กชายธาวิน ศรีบุญเรือง	ชั้น ม.2/1 เลขที่ 10
เด็กชายพัฒนพล สุธรรม	ชั้น ม.2/1 เลขที่ 19
เด็กชายภาณุวัฒน์ เทียนทอง	ชั้น ม.2/1 เลขที่ 21

โรงเรียนปิริยาลัยจังหวัดแพร่ ตำบลในเวียง อำเภอเมือง จังหวัดแพร่

สำนักงานเขตพื้นที่การศึกษามัธยมศึกษาแพร่
สำนักงานคณะกรรมการการศึกษาขั้นพื้นฐาน
กระทรวงศึกษาธิการ

โครงการ สาขา วิทยาศาสตร์สิ่งแวดล้อม
ระบบเลี้ยงกุ้งฝอย IoT ด้วยบอร์ด ESP32

โดย

เด็กชายธาวิน ศรีบุญเรือง	ชั้น	ม.2/1 เลขที่ 10
เด็กชายพัฒนพล สุธรรม	ชั้น	ม.2/1 เลขที่ 19
เด็กชายภาณุวัฒน์ เทียนทอง	ชั้น	ม.2/1 เลขที่ 21

ครูที่ปรึกษา

นายศุภณัฐ พัฒนสารินทร์

โครงการนี้เป็นส่วนหนึ่งของการเรียนในรายวิชาวิทยาศาสตร์โลกทั้งระบบ ปีการศึกษา 2567
ตามหลักสูตร สถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี (สสวท.)

ชื่อโครงการ	ระบบเลี้ยงกุ้งฝอย IoT ด้วยบอร์ด ESP32				
ผู้จัดทำโครงการ	1. เด็กชายธาวิน	ศรีบุญเรือง	ชั้น	ม.2/1	เลขที่ 10
	2. เด็กชายพัฒนพล	สุธรรม	ชั้น	ม.2/1	เลขที่ 19
	3. เด็กชายภาณุวัฒน์	เทียนทอง	ชั้น	ม.2/1	เลขที่ 21
โรงเรียน	ปริยาลัยจังหวัดแพร่				
ครูที่ปรึกษา	นายศุภณัฐ พัฒนสารินทร์				

บทคัดย่อ

โครงการ ระบบเลี้ยงกุ้งฝอย IoT ด้วยบอร์ด ESP32 จัดทำขึ้นเพื่อศึกษาการพัฒนา ระบบเลี้ยงกุ้งฝอย ด้วยเทคโนโลยี IoT และทดสอบประสิทธิภาพการทำงาน โดยใช้บอร์ด ESP32 ควบคุมอุปกรณ์ 4 ชนิด ได้แก่ เครื่องให้ออกซิเจน เครื่องกรองน้ำ เครื่องระบายความร้อน และเครื่องทำความร้อน เพื่อสร้างสภาพแวดล้อมที่เหมาะสมสำหรับการเลี้ยงกุ้งฝอย การทดลองแบ่งเป็น 2 ตอน ตอนแรกคือการพัฒนา ระบบควบคุม ตอนที่สอง คือการทดสอบระบบโดยเลี้ยงไว้ 3 วัน และเลี้ยง 2 แบบ คือ ใช้ปุ๋ยมูลไก่ กรวด และทรายในบ่อเลี้ยง และแบบ ไม่ใช้ปุ๋ยมูลไก่ กรวด และทรายในบ่อเลี้ยง

จากการทดลองพบว่า ระบบเลี้ยงกุ้งฝอย IoT ด้วยบอร์ด ESP32 สามารถควบคุมอุปกรณ์ให้ทำงานได้อย่างสม่ำเสมอ การเลี้ยงแบบที่ 1 ผลจากการทดลองแสดงให้เห็นว่าอัตราการรอดอยู่ที่ 25% และแบบที่ 2 พบว่าอัตราการรอดสูงขึ้นถึง 100%

สรุปได้ว่าระบบเลี้ยงกุ้งฝอย IoT ด้วยบอร์ด ESP32 สามารถควบคุมอุปกรณ์ให้ทำงานได้อย่างมีประสิทธิภาพ และการเลี้ยงแบบไม่ใช้ปุ๋ยมูลไก่ กรวด และทราย มีอัตราการรอดของกุ้งฝอยที่ดีที่สุด

กิตติกรรมประกาศ

โครงการวิทยาศาสตร์ เรื่องระบบเลี้ยงกุ้งฝอย IoT ด้วยบอร์ด ESP32 จัดทำขึ้นเพื่อศึกษาเกี่ยวกับการทำระบบเลี้ยงกุ้งฝอย โดยได้รับการสนับสนุนจาก คุณครูศุภณัฐ พัฒนสารินทร์ คุณครูที่ปรึกษาที่ได้ให้คำแนะนำปรึกษา และให้ข้อมูลต่างๆ ขอขอบพระคุณที่ได้ให้คำปรึกษา

ขอขอบพระคุณ คุณครูศุภณัฐ พัฒนสารินทร์ ที่ให้คำแนะนำและเอื้อเฟื้อวัสดุอุปกรณ์ต่างๆ สำหรับการทำโครงการ

ขอขอบพระคุณ คุณครูอำนาจ แก้วแดง ที่ให้คำแนะนำในการออกแบบสำหรับการทำโครงการ

อนึ่ง คณะผู้จัดทำหวังว่า โครงการวิทยาศาสตร์ เรื่อง ระบบเลี้ยงกุ้งฝอย IoT ด้วยบอร์ด ESP32 จะมีประโยชน์อยู่ไม่น้อย จึงขอมอบส่วนดีทั้งหมดนี้ให้แก่เหล่าครูบาอาจารย์ที่ได้ประสิทธิ์ประสาทวิชาจนทำให้ผลงานวิจัยเป็นประโยชน์ต่อผู้เกี่ยวข้องและขอมอบความกตัญญูทเวทิตาคุณแด่บิดา มารดา และผู้มีพระคุณทุกท่านตลอดจนเพื่อนๆ ที่คอยให้ความช่วยเหลือและกำลังใจ

คณะผู้จัดทำ

18 กุมภาพันธ์ 2568

สารบัญ

	หน้า
บทคัดย่อ	ก
กิตติกรรมประกาศ	ข
สารบัญ	ค
สารบัญรูป	ง
สารบัญตาราง	จ
บทที่ 1 บทนำ	1
บทที่ 2 เอกสารที่เกี่ยวข้อง	4
บทที่ 3 อุปกรณ์และวิธีการดำเนินการ	19
บทที่ 4 ผลการดำเนินงาน	23
บทที่ 5 สรุปผลการดำเนินงาน	25
บรรณานุกรม	27
ภาคผนวก	29

สารบัญรูป

	หน้า
รูป 1 กุ้งฝอย	5
รูป 2 ไมโครคอนโทรลเลอร์ ESP32 DEVKIT V1	8
รูป 3 เซ็นเซอร์วัดอุณหภูมิ DS18B20	9
รูป 4 TDS Sensor	10
รูป 5 DS1302 RTC Module	11
รูป 6 ฐานเครื่องให้อาหาร	16
รูป 7 ที่ใส่อาหาร	16
รูป 8 เฟืองปิด-เปิดช่องใส่อาหาร	16
รูป 9 เครื่องให้ออกซิเจน	17
รูป 10 เครื่องกรองน้ำ	17
รูป 11 คอยล์เย็น	18
รูป 12 เครื่องทำความร้อน	18
รูป 13 ออกแบบระบบเลี้ยงกุ้งฝอย	20
รูป 14 ระบบเลี้ยงกุ้งฝอย IoT	21
รูป 15 ทดสอบประสิทธิภาพด้วยการเตรียมบ่อที่มีปุ๋ยมูลไก่ กรวดและทราย	24
รูป 16 ทดสอบประสิทธิภาพด้วยการเตรียมบ่อที่ไม่มีปุ๋ยมูลไก่ กรวดและทราย	24

สารบัญตาราง

	หน้า
ตาราง 1 ประสิทธิภาพของระบบเลี้ยงกุ้งฝอย IoT ด้วยการเตรียมบ่อที่มีปุ๋ยมูลไก่ กรวดและทราย	23
ตาราง 2 ประสิทธิภาพของระบบเลี้ยงกุ้งฝอย IoT ด้วยการเตรียมบ่อที่ไม่มีปุ๋ยมูลไก่ กรวดและทราย	24

บทที่ 1

บทนำ

ที่มาและความสำคัญ

กุ้งฝอยเป็นสัตว์น้ำเศรษฐกิจที่มีความต้องการสูงในตลาด เนื่องจากใช้เป็นวัตถุดิบในการปรุงอาหาร และเลี้ยงปลาในเชิงพาณิชย์ แต่การเลี้ยงกุ้งฝอยในปัจจุบันยังคงพึ่งพาวิธีการแบบทั่วไปที่ต้องใช้แรงงานในการดูแลบ่อเลี้ยง เช่น การตรวจสอบคุณภาพน้ำ การให้อาหาร และการเฝ้าระวังปัญหาในบ่อเลี้ยง ซึ่งใช้เวลาจำนวนมาก นอกจากนี้ สำหรับพื้นที่เลี้ยงขนาดเล็ก เช่น ในครัวเรือนหรือในชุมชนเมือง การเลี้ยงกุ้งฝอยมักประสบปัญหาด้านการจัดการพื้นที่และความสม่ำเสมอของคุณภาพน้ำ ทำให้ผลผลิตไม่เป็นไปตามเป้าหมายในบางสภาพแวดล้อม

ด้วยความก้าวหน้าของเทคโนโลยี Internet of Things (IoT) และอุปกรณ์ควบคุม เช่น บอร์ด อีเอสพี32 (ESP32) ทำให้สามารถนำมาใช้พัฒนาเป็นระบบเลี้ยงกุ้งฝอยแบบอัตโนมัติที่มีความแม่นยำในการตรวจสอบและควบคุมปัจจัยแวดล้อมในบ่อเลี้ยงได้อย่างมีประสิทธิภาพ

จากความสำคัญและปัญหาข้างต้น คณะผู้จัดทำโครงการจึงสร้างระบบเลี้ยงกุ้งฝอยที่ใช้เทคโนโลยี IoT เพื่อแก้ปัญหาที่เกิดขึ้นจากการเลี้ยงแบบทั่วไปในปัจจุบัน โดยระบบนี้จะช่วยเลี้ยงและดูแลกุ้งฝอย ลดความเสี่ยงจากปัจจัยแวดล้อมที่ไม่เหมาะสม นอกจากนี้ยังเก็บข้อมูลของคุณภาพน้ำต่างๆ รวมถึงแจ้งเตือนเมื่อปัจจัยสภาพแวดล้อมไม่เหมาะสม และเพิ่มความสะดวกสบายให้กับผู้เลี้ยงในทุกระดับ

วัตถุประสงค์

1. เพื่อศึกษาหลักการและสร้างระบบเลี้ยงกุ้งฝอย IoT ด้วยบอร์ด ESP32
2. เพื่อทดสอบประสิทธิภาพการทำงานของระบบเลี้ยงกุ้งฝอย IoT ด้วยบอร์ด ESP32

ตัวแปร

ตอนที่ 1 ศึกษาการสร้างระบบเลี้ยงกุ้งฝอยโดยใช้เทคโนโลยี IoT

ตัวแปรต้น : หลักการและวิธีการทำงานของระบบเลี้ยงกุ้งฝอย

ตัวแปรตาม : ระบบเลี้ยงกุ้งฝอยแบบ IoT

ตัวแปรควบคุม : เซนเซอร์วัดอุณหภูมิ, เซนเซอร์วัดค่า Total dissolved solids (TDS), ปีมล, เครื่องกรองน้ำ, เครื่องระบายความร้อน, เครื่องทำความร้อน, เครื่องให้อาหาร, ไมโครคอนโทรลเลอร์, ขนาดของตู้กระจก

ตอนที่ 2 ทดสอบประสิทธิภาพการทำงานของระบบเลี้ยงกุ้งฝอยโดยใช้เทคโนโลยี IoT

ตัวแปรต้น : การเลี้ยงกุ้งฝอย, การเตรียมบ่อเลี้ยงกุ้งฝอย

ตัวแปรตาม : อัตราการรอดของกุ้งฝอย

ตัวแปรควบคุม : ระบบเลี้ยงกุ้งฝอย IoT, ขนาดตู้กระจก, แหล่งน้ำที่ใช้ในการเลี้ยง, ปริมาณและประเภทของอาหารที่ใช้, จำนวนและชนิดของกุ้งฝอย

นิยามศัพท์เฉพาะ

ระบบเลี้ยงกุ้งฝอย IoT โดยใช้บอร์ด ESP32 หมายถึง ระบบที่สร้างขึ้นมาเพื่อควบคุมสภาพแวดล้อมต่างๆในการเลี้ยงกุ้งฝอย โดยใช้เทคโนโลยี IoT ร่วมกับบอร์ดไมโครคอนโทรลเลอร์ ESP32 สำหรับการประมวลผลข้อมูลและการควบคุมอุปกรณ์

เทคโนโลยี Internet of Things (IoT) คือเทคโนโลยีที่เชื่อมต่ออุปกรณ์ต่างๆเข้ากับระบบอินเทอร์เน็ตเพื่อให้สามารถส่งข้อมูลระหว่างกันและควบคุมจากระยะไกล

อัตราการรอดของกุ้งฝอยคือสัดส่วนหรือเปอร์เซ็นต์ของจำนวนกุ้งฝอยที่ยังมีชีวิตรอดอยู่เมื่อสิ้นสุดการทดลอง เทียบกับจำนวนกุ้งฝอยเริ่มต้นที่นำมาเลี้ยงในระบบ

การแจ้งเตือนผ่าน Telegram คือการส่งข้อความหรือข้อมูลแจ้งเตือนไปยังผู้ใช้งานผ่านแอปพลิเคชัน Telegram โดยอาศัยเทคโนโลยีการเชื่อมต่อระหว่างระบบควบคุม กับ Telegram Bot API เพื่อส่งข้อมูลในรูปแบบข้อความ รูปภาพ หรือข้อมูลอื่นๆ ในเวลาจริง (real-time) เพื่อแจ้งเตือนเหตุการณ์หรือสถานะที่สำคัญ เช่น การเปลี่ยนแปลงของคุณภาพน้ำ การทำงานของอุปกรณ์ หรือปัญหาในระบบเลี้ยงกุ้งฝอย

การเก็บข้อมูลผ่าน Google Sheets คือการบันทึกและจัดเก็บข้อมูลจากระบบหรืออุปกรณ์ต่างๆ ลงใน Google Sheets ซึ่งเป็นแอปพลิเคชันสำหรับการจัดการสเปรดชีตออนไลน์ของ Google โดยใช้ Google Sheets API หรือบริการเชื่อมต่ออื่นๆ ข้อมูลที่เก็บอาจเป็นค่าที่วัดได้จากเซนเซอร์ เช่น อุณหภูมิ หรือค่า TDS และสามารถนำไปใช้วิเคราะห์หรือแสดงผลในรูปแบบที่ต้องการได้ ทั้งนี้ Google Sheets ยังรองรับการทำงานร่วมกันและเข้าถึงข้อมูลได้จากทุกที่ผ่านอินเทอร์เน็ต

ขอบเขตของการศึกษา

1. ฮาร์ดแวร์และการออกแบบระบบ:

1. ใช้บอร์ด ESP32 เป็นหน่วยควบคุมหลัก
2. ใช้เซ็นเซอร์วัดอุณหภูมิ (DS18B20) และเซ็นเซอร์วัดปริมาณสารละลายในน้ำ (TDS Sensor) เป็นเซ็นเซอร์สำหรับวัดปัจจัยแวดล้อม
3. ใช้ปั๊มลม เครื่องระบายความร้อน เครื่องทำความร้อน และเครื่องกรองน้ำเป็นเครื่องควบคุมปัจจัยในการเลี้ยง
4. อุปกรณ์ให้อาหารควบคุมด้วยมอเตอร์เซอร์โว
5. จอ OLED สำหรับแสดงค่าที่วัดได้

2. ซอฟต์แวร์และการเชื่อมต่อ IOT:

1. การพัฒนาโปรแกรมบน ESP32 ด้วย Arduino IDE
2. ใช้ Google Sheet ในการเก็บข้อมูลอุณหภูมิ ค่าสารที่ละลายในน้ำ และการเปิดปิดของอุปกรณ์ต่างๆ
3. ใช้การแจ้งเตือนผ่าน Telegram หากว่ามีความผิดปกติ

ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ระบบเลี้ยงกุ้งฝอยที่ใช้เทคโนโลยี IoT ในการเลี้ยง ที่มีการควบคุมปัจจัยต่างๆที่จำเป็นในการเลี้ยงกุ้งฝอย
2. ได้รับความรู้จากการประยุกต์ใช้เทคโนโลยี IoT ในการนำมาเลี้ยงกุ้งฝอย

บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

การทำโครงการเรื่อง ระบบเลี้ยงกุ้งฝอย IoT คณะผู้จัดทำได้ศึกษาแนวคิดและเอกสารที่เกี่ยวข้องโดยนำเสนอตามลำดับ ดังนี้

1. Internet of Things หรือ IoT คืออะไร

มีทนา วิบูลย์ยศศักดิ์ (ม.ป.ป.) ได้ให้ความหมายของ Internet of Things (IoT) ไว้ว่าเป็นการที่อุปกรณ์อิเล็กทรอนิกส์ต่างๆ สามารถเชื่อมโยงหรือส่งข้อมูลถึงกันได้ด้วยอินเทอร์เน็ต โดยไม่ต้องป้อนข้อมูล การเชื่อมโยงนี้ช่วยให้เราสามารถสั่งการควบคุมการใช้งานอุปกรณ์อิเล็กทรอนิกส์ต่างๆ ผ่านทางเครือข่ายอินเทอร์เน็ตได้ ไปจนถึงการเชื่อมโยงการใช้งานอุปกรณ์อิเล็กทรอนิกส์ต่างๆ ผ่านทางเครือข่ายอินเทอร์เน็ตเข้ากับการใช้งานอื่นๆ จนเกิดเป็นบรรดา Smart ต่างๆ ได้แก่ Smart Device, Smart Grid, Smart Home, Smart Network, Smart Intelligent Transportation ทั้งหลายที่เราเคยได้ยินนั่นเอง ซึ่งแตกต่างจากในอดีตที่อุปกรณ์อิเล็กทรอนิกส์เป็นเพียงสื่อกลางในการส่งและแสดงข้อมูลเท่านั้น

กล่าวได้ว่า Internet of Things นี้ได้แก่การเชื่อมโยงของอุปกรณ์อัจฉริยะทั้งหลายผ่านอินเทอร์เน็ตที่เราคุ้นเคย เช่น แอปพลิเคชัน แวนตาไกลกลาส รองเท้าวิ่งที่สามารถเชื่อมต่อข้อมูลการวิ่ง ทั้งความเร็ว ระยะทาง สถานที่ และสถิติได้

นอกจากนั้น Cloud Storage หรือ บริการรับฝากไฟล์และประมวลผลข้อมูลของคุณผ่านทางออนไลน์ หรือเราเรียกอีกอย่างว่า แคล้งเก็บข้อมูลบนก้อนเมฆ เป็นอีกสิ่งหนึ่งที่เรากำลังใช้อยู่แต่ไม่รู้ว่าเป็นหนึ่งในรูปแบบของ Internet of Things สมัยนี้ผู้ใช้นิยมเก็บข้อมูลไว้ในก้อนเมฆมากขึ้น เนื่องจากมีข้อดีหลายประการ คือ ไม่ต้องกลัวข้อมูลสูญหายหรือถูกโจรกรรม ทั้งยังสามารถกำหนดให้เป็นแบบส่วนตัวหรือสาธารณะก็ได้ เข้าถึงข้อมูลได้ทุกที่ทุกเวลาด้วยอุปกรณ์อิเล็กทรอนิกส์ใดๆ ผ่านเครือข่ายอินเทอร์เน็ต แถมยังมีพื้นที่ใช้สอยมาก มีให้เลือกหลากหลาย ช่วยเราประหยัดค่าใช้จ่ายได้อีกด้วย เนื่องจากเราไม่ต้องเสียเงินซื้ออุปกรณ์จัดเก็บข้อมูล เช่น ฮาร์ดไดรฟ์ หรือ Flash drive ต่างๆ

เทคโนโลยี Internet of Things มีประโยชน์ในหลายด้านทั้งเรื่องการเก็บข้อมูลที่แม่นยำและเป็นปัจจุบัน ช่วยลดต้นทุน แถมยังช่วยเพิ่มผลผลิตของพนักงานหรือผู้ใช้งานได้ แม้ว่าแนวโน้มของ IoT มีแต่จะเพิ่มขึ้นด้วยคุณภาพประโยชน์ตามที่ได้กล่าวมาแล้ว แต่ประโยชน์ใดๆ นั้นก็มาพร้อมกับความเสี่ยง เพราะความท้าทายในการรักษาความปลอดภัยของเครือข่ายใหม่ที่เกิดขึ้นนั้น จะผลักดันให้ผู้เชี่ยวชาญมีการรับมือทางด้านความปลอดภัยมากขึ้น ในทางตรงกันข้ามแฮกเกอร์หรือผู้ไม่หวังดีก็ทำงานหนักเพื่อที่จะเข้าควบคุม โจมตีเครือข่าย หรือเรียกค่าไถ่ในช่องโหว่ที่ IoT มีอยู่ ฉะนั้นผู้เชี่ยวชาญด้านความปลอดภัยทาง IoT จึงจำเป็นต้องพัฒนามาตรการ และระบบรักษาความปลอดภัยไอทีควบคู่กันไป เพื่อให้ธุรกิจและการใช้งาน IoT สามารถขับเคลื่อนต่อไปได้

2. กุ้งฝอย



รูป 1 กุ้งฝอย

(ที่มา: gosmartfarmer, 2567)

บัญชา ทองมี (2555) ได้ให้ข้อมูลของกุ้งฝอยดังรูป 1 ไว้ว่าชื่อสามัญของกุ้งฝอยคือ Riceland Prawn หรือ Lanchester's Freshwater Prawn ชื่อวิทยาศาสตร์ว่า *Macrobrachium lanchesteri* มีขนาดความยาวสูงสุดประมาณ 5 ซม. ลูกกุ้งฝอยมีลักษณะคล้ายลูกกุ้งก้ามกรามมาก โดยหนามด้านบนมี 4-7 ซี่ และด้านล่าง 1-2 ซี่ กุ้งฝอยตัวเต็มวัยมีลักษณะลำตัวใส ช่วยในการพรางตัวจากศัตรูธรรมชาติ บริเวณหัวมีหนวดคู่สั้นและคู่ยาวอย่างละ 1 คู่ โดยหนวดคู่ยาวสามารถรับสัมผัสและกลิ่นได้ดี ใช้เตือนภัยและหาอาหาร กุ้งฝอยมีขาเดิน 5 คู่ บริเวณด้านล่างของส่วนอก ขาเดินคู่ที่ 1 และ 2 มีลักษณะเป็นก้ามหนีบ โดยก้ามคู่ที่ 2 มีขนาดใหญ่กว่า ใช้สำหรับจับอาหาร ลำตัวของกุ้งแบ่งเป็นปล้องทั้งหมด 6 ปล้อง บริเวณด้านล่างของปล้องที่ 1-5 มีขาว่ายน้ำปล้องละ 1 คู่ ในกุ้งเพศเมียจะเก็บไข่ไว้บริเวณขาว่ายน้ำของปล้องที่ 4 ไข่กุ้งฝอย เป็นชนิดไข่จมติด มีเมือกเหนียวห่อหุ้ม ลักษณะคล้ายไข่ไก่ ขนาดประมาณ 0.5 มม. ไข่หนึ่งฟองหนักประมาณ 0.000059 กรัม แม่กุ้งที่มีความยาว 1.8-2.5 ซม. จะมีไข่เฉลี่ย 200-250 ฟอง ในระหว่างการวางไข่ ไข่จะถูกปล่อยออกจากขาเดินคู่ที่สาม และเคลื่อนผ่านขาเดินคู่ที่ห้า บริเวณดังกล่าวไข่จะได้รับการผสมกับน้ำเชื้อของเพศผู้ หลังจากการปฏิสนธิในเวลา 2-3 ชั่วโมง ไข่จะเปลี่ยนสีจากสีเขียวเข้มเป็นสีเขียวย่อนใน 3 วัน และเปลี่ยนเป็นสีเหลืองใน 9 วัน จากนั้นไข่จะเปลี่ยนเป็นสีเทาใส และเห็นตาของตัวอ่อนเป็นจุดดำสองจุดชัดเจนไข่จะฟักเป็นตัวเมื่ออายุ 21-25 วัน ลูกกุ้งแรกเกิดจะลอยตัวในน้ำโดยหัวห้อยลง หลังจาก 25-30 วัน ลูกกุ้งจะเริ่มเคลื่อนที่โดยใช้ขาว่ายน้ำในลักษณะคล้ายกับพ่อแม่พันธุ์

กุ้งฝอยอาศัยอยู่ในแหล่งน้ำจืด มักชอบซ่อนตัวใต้ก้อนหินหรือเกาะตามพรรณไม้ น้ำ โดยพบได้ในน้ำไหลเอื่อยหรือน้ำนิ่งที่มีลักษณะขุ่น และมีระดับน้ำลึกไม่เกิน 1 เมตร บริเวณที่มีอินทรีย์วัตถุทับถมกัน น้ำมีคุณภาพค่อนข้างดี

กุ้งฝอยกินอาหารได้ทั้งพืชและสัตว์ โดยเฉพาะของเน่าเปื่อยซึ่งเป็นอาหารที่ชื่นชอบ กุ้งฝอยมักออกหาอาหารในเวลากลางคืน แต่สามารถกินอาหารในเวลากลางวันได้ หากปริมาณออกซิเจนในน้ำต่ำ ความสามารถในการกินอาหารของกุ้งฝอยจะลดลง

การเตรียมบ่อดิน: สูบน้ำออกจากบ่อโรยปูนขาว 60-120 กก./ไร่ ตากบ่อให้แห้งใช้เวลา 2-3 สัปดาห์ กั้นคอกล้อมรอบคันบ่อ ด้วยอวนสีฟ้าหรือตาข่ายตาละเอียดเพื่อป้องกันศัตรูธรรมชาติ เช่น ปลา กบ เขียดและแมลงน้ำเช่นตัวอ่อนแมลงปอ มวนกรรเชียง จากนั้นเติมน้ำใส่บ่อสูง 40-50 ซม. กรองน้ำที่นำเข้าด้วยผ้ากรองตาถี่เพื่อป้องกันไข่ปลาและลูกปลา ขนาดเล็กเล็ดลอดเข้าบ่อ ใส่ปุ๋ยมูลไก่อัตรา 60-120 กิโลกรัม/ไร่/สัปดาห์ ทิ้งไว้ 3-4 วันจนสีน้ำเริ่มเขียว เติมน้ำจนได้ ระดับความสูงประมาณ 1 เมตร

วิธีการเพาะฟักลูกกุ้ง: รวบรวมและคัดเลือกแม่กุ้งไข่แก่สีเทาใส (ระยะมีตา) จากบ่อดิน กระชังหรือจากแหล่งน้ำธรรมชาติ จำนวน 80-100 ตัว ใส่ในตะแกรงพลาสติกที่แขวนไว้ในกระชังฟักไข่ผ้าโอลอนแก้ว ขนาด 1 x 1 x 1 ลบ.ม.ที่เตรียมไว้ในบ่อ ซีเมนต์หรือบ่อดิน ให้อาหารเม็ดสำเร็จรูประดับโปรตีน 35% แก่แม่กุ้งฝอย โดยใส่ในภาชนะรูปจานยกขอบสูงป้องกัน อาหารตกลงในกระชังฟักไข่ แบ่งให้อาหาร 2 ครั้งเช้าและเย็น ประมาณ 3-5 วันไข่จะฟักออกเป็นตัวคัดแยกแม่กุ้งที่ วางไข่แล้วออกไปรวบรวมลูกกุ้งที่มีอายุรุ่นเดียวกันหรืออายุที่แตกต่างไม่เกิน 3 วันไปไว้ในกระชังอนุบาลคอกที่กั้นด้วยอวนสีฟ้าหรือกระชังอวนสีฟ้าที่กางไว้ในบ่อดิน

ศุภวิชญ์ ขวลิขิตพิเชฐ (2561) ได้ทำการวิจัยไว้ว่าค่าความเป็นกรด-ด่างของน้ำจะเปลี่ยนแปลงตลอดเวลาเนื่องจากการสะสมของเสียที่กุ้งถ่ายออกมา รวมทั้งเศษอาหารตกค้างในบ่อและซากพืชซากสัตว์ การสังเคราะห์แสงของพืชน้ำก็มีส่วนทำให้ค่าความเป็นกรด-ด่างเปลี่ยนแปลง ในช่วงบ่ายเวลาประมาณ 15.00-16.00 น. การสังเคราะห์แสงจะถึงจุดสูงสุด ทำให้ค่าความเป็นกรด-ด่างในน้ำสูงมาก แต่ในช่วงกลางคืนพืชน้ำต่าง ๆ จะหยุดสังเคราะห์แสงเนื่องจากไม่มีแสงแดด มีแต่การหายใจเพียงอย่างเดียว ค่าความเป็นกรด-ด่างจะลดลงมาก ค่าความเป็นกรด-ด่างในบ่อกุ้งควรรักษาให้อยู่ในระดับ 7-9 และระดับที่เหมาะสมที่สุดคือ 7.5-8.5 ซึ่งเหมาะสมต่อการเจริญเติบโตของกุ้งมากที่สุด กุ้งสามารถจะกินอาหารได้ดีด้วย ดังนั้นควรรักษา ระดับค่าความเป็นกรด-ด่างให้อยู่ในช่วง 7.5-8.5 ถ้าสูงหรือต่ำกว่าควรปรับแก้ไข

ปริมาณออกซิเจนที่ละลายในน้ำจะลดลงเนื่องจากการใช้หายใจของกุ้งและสิ่งมีชีวิตอื่นๆ ในน้ำ ซึ่งทำให้สภาพแวดล้อมเปลี่ยนแปลงไป ค่าออกซิเจนที่ละลายในน้ำจะต่ำที่สุดในช่วงเวลา 02.00-05.00 น. ก่อนพระอาทิตย์ขึ้น ค่าต่ำสุดที่กุ้งจะทนได้คือประมาณ 0.35-0.9 พีพีเอ็ม สำหรับการเลี้ยงแบบหนาแน่นในสภาพน้ำที่เขียว ควรรักษาระดับออกซิเจนที่ละลายในน้ำให้สูงกว่า 3 พีพีเอ็ม ถ้าออกซิเจนละลายในน้ำไม่พอ กุ้งจะว่ายบนผิวน้ำ มีอาการลอยและเกาะตามตลิ่ง จะไม่ว่ายลงไปใต้น้ำแม้จะมีคนเดินมาใกล้ก็ตาม ควรรีบเปลี่ยนถ่ายน้ำหรือเปิดเครื่องตีผิวน้ำให้เต็มที่ทันทีเพื่อเพิ่มค่าออกซิเจนในน้ำ

อุณหภูมิของน้ำที่เหมาะสมที่สุดในช่วง 25-30 องศาเซลเซียส ถ้าอุณหภูมิสูงหรือต่ำเกินไป อาจทำให้กุ้งอ่อนแอหรือตายได้

3.สาหร่ายหางกระรอก

สาหร่ายหางกระรอกมีชื่อวิทยาศาสตร์คือ *Hydrilla verticillate* อยู่ในวงศ์ Hydrocharitaceae (วงศ์สั่นตะวา) สกุลมีชื่อว่า Hydrilla และชื่อสามัญคือ Florida elodea

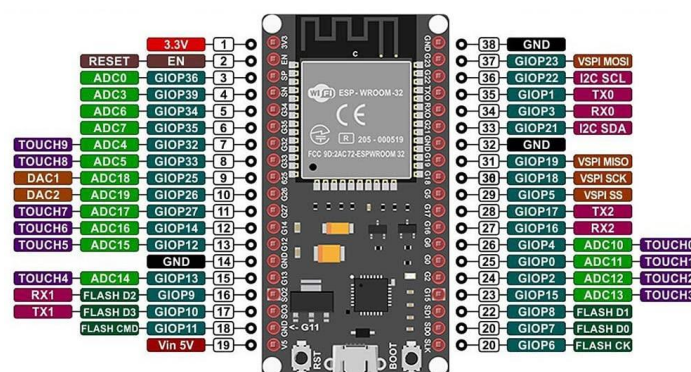
สาหร่ายหางกระรอก (*Hydrilla verticillata*) นั้น เดิมทีเป็นวัชพืชน้ำที่พบในแหล่งน้ำจืด เช่น ห้วย หนอง คลอง และบึง โดยเจริญเติบโตในพื้นที่ที่มีแสงแดดและน้ำใสสะอาด สันนิษฐานว่ามีถิ่นกำเนิดในพื้นที่เขตอบอุ่นและเขตร้อนของซีกโลกตะวันออก รวมถึงเป็นพืชพื้นเมืองในประเทศไทย ต่อมามนุษย์ได้นำสาหร่ายชนิดนี้มาใช้ประโยชน์ในฐานะไม้ประดับสำหรับตู้ปลา เนื่องจากมีความสวยงามและสามารถปลูกได้ง่าย ลำต้นมีลักษณะเรียวยาว ทอดตามความลึกของน้ำ มีความยาวได้ถึง 3 เมตร ใบจมอยู่ใต้น้ำ ลักษณะเรียวยาวแบนคล้ายรูปกระบี่ ขอบหยักเล็กน้อย ใบยาวประมาณ 10–20 มิลลิเมตร และกว้างประมาณ 2–5 มิลลิเมตร โดยใบจะแตกออกเป็นวงรอบข้อตามลำต้น ไม่มีการเชื่อมต่อกัน ใบ ดอกเป็นดอกเดี่ยวที่ช่อใบ ดอกตูมอยู่ใต้น้ำ และเมื่อบานจะลอยเหนือผิวน้ำ มีสีขาวประกอบด้วย 3 กลีบเลี้ยง 3 กลีบดอก และเกสรเพศผู้ 3 ยอด รวมถึงเกสรเพศเมีย 1 ยอด

ประโยชน์ของสาหร่ายหางกระรอกคือ เพิ่มออกซิเจนในน้ำ เพราะสาหร่ายหางกระรอกช่วยเพิ่มออกซิเจนในน้ำผ่านกระบวนการสังเคราะห์แสง ซึ่งมีประโยชน์ต่อการเจริญเติบโตของสัตว์น้ำ ใบของสาหร่ายเป็นแหล่งอาหารที่ดีสำหรับปู กุ้ง และปลา โดยอุดมไปด้วยวิตามินบี 12 เหล็ก และแคลเซียม อีกทั้งยังเป็นที่หลบภัยสำหรับสัตว์น้ำได้ เพราะซอสาหร่ายสามารถเป็นที่หลบภัยของสัตว์น้ำตัวเล็ก เช่น กุ้งฝอย และสัตว์อื่นๆ

สาหร่ายหางกระรอกเป็นพืชน้ำที่แพร่กระจายได้ง่าย ไม่ต้องการการดูแลซับซ้อน การขยายพันธุ์สามารถทำได้โดยการนำต้นพันธุ์แช่ในน้ำที่ไม่มีคลอรีน เช่น น้ำบาดาล หรือน้ำจากแหล่งธรรมชาติ และควรปลูกในพื้นที่ที่มีแสงแดดรำไร การปลูกในตู้ปลาสามารถทำได้โดยนำต้นพันธุ์สาหร่ายมาแช่ในน้ำสะอาดโดยไม่ต้องใช้ดิน จัดแสงไฟให้เหมาะสม (เช่น ใช้หลอดฟลูออเรสเซนต์กำลังไฟ 1 วัตต์ต่อน้ำ 1 ลิตร เปิดขั้นต่ำ 4 ชั่วโมงต่อวัน) เพื่อให้สาหร่ายสังเคราะห์แสงได้ดี แต่ข้อควรระวังคือ ให้หลีกเลี่ยงการปลูกในน้ำที่มีคลอรีน แต่ถ้าหากไม่มีแสงธรรมชาติ ควรใช้ไฟสังเคราะห์แสงเพื่อทดแทน

สาหร่ายหางกระรอกจึงเป็นพืชน้ำที่มีประโยชน์หลายประการ ทั้งในด้านการตกแต่งตู้ปลาและการเสริมสร้างความสมดุลของระบบนิเวศในน้ำ (plantlover, ม.ป.ป.)

4. ไมโครคอนโทรลเลอร์ ESP32



รูป 2 ไมโครคอนโทรลเลอร์ ESP32 DEVKIT V1

(ที่มา: GROBOTRONICS, ม.ป.ป.)

ศุภรัตน์ แยมครวญ (2566) ได้อธิบายบอร์ด อีเอสพี32 (ESP32) ดังรูป 2 ไว้ว่าเป็นบอร์ด ไมโครคอนโทรลเลอร์ราคาไม่สูง ใช้พลังงานต่ำ และพร้อมกับการเชื่อมต่อ Wi-Fi และ Bluetooth ซึ่งเป็นการ พัฒนาจากรุ่นก่อนหน้า ESP8266 โดยใช้ระบบ System-on-chip (SoC) ที่ถูกพัฒนาจาก Espressif Systems

ด้วยองค์ประกอบของ ESP32 ที่มาพร้อมกับสายอากาศในตัวเอง (Antenna Integration), RF balun, การขยายกำลังส่ง, การขยายสัญญาณรบกวนต่ำ, ตัวกรอง, และโมดูลการจัดการพลังงาน โดยทั้งหมดนี้จะใช้พื้นที่บนบอร์ดวงจรพืมน้อยมาก โดยบอร์ด ESP32 นี้ใช้กับชิป Wi-Fi และ Bluetooth แบบคู่ 2.4 กิกะเฮิร์ต และมีเทคโนโลยีการผลิตที่มีพลังงานต่ำของ TSMC ที่ 40 นาโนเมตร ที่มีคุณสมบัติทาง พลังงานที่ใช้ต่ำ และมีระบบ RF ซึ่งเป็นเทคโนโลยีที่น่าเชื่อถือ และสามารถปรับขนาดใช้งานได้หลากหลายใน แอปพลิเคชันต่างๆ โดยเฉพาะด้านอินเทอร์เน็ตของสรรพสิ่ง (IoT) ESP32 เป็นไมโครคอนโทรลเลอร์ที่ได้รับความนิยมสูงในการพัฒนาอุปกรณ์ IoT เนื่องจากมีคุณสมบัติที่มีประสิทธิภาพด้านพลังงาน และด้วย ความสามารถในการเชื่อมต่อ Wi-Fi Bluetooth อย่างคล่องตัว ทำให้นักพัฒนาสามารถเชื่อมต่อไปยัง เครือข่ายไร้สาย สื่อสารกับอุปกรณ์อื่นๆ ได้อย่างง่ายดาย

5. เซ็นเซอร์วัดอุณหภูมิ DS18B20



รูป 3 เซ็นเซอร์วัดอุณหภูมิ DS18B20

(ที่มา: IOT.CodeMobiles, ม.ป.ป.)

เซ็นเซอร์ DS18B20 ดังรูป 3 ผลิตโดยบริษัท Dallas Semiconductor แต่ปัจจุบันถูกบริษัท Maxim integrated ซื้อมาผลิตให้การใช้งานนั้นมีขาใช้งานเพียง 3 ขา มีสายสัญญาณต่อเข้าเพียงเส้นเดียว ที่เหลืออีกสองขาคือไฟเลี้ยง DS18B20 ใช้ไฟเลี้ยง 3 ถึง 5.5 โวลต์ (V) กระแสไฟฟ้าที่ใช้ประมาณ 1 มิลลิแอมป์ (mA) ช่วงอุณหภูมิที่วัดได้จะอยู่ในช่วง -55 ถึง 125 องศาเซลเซียส ความคลาดเคลื่อนประมาณ ± 0.5 องศาเซลเซียส

การใช้งาน DS18B20 กับ Arduino นั้นสามารถใช้งานได้ง่าย มีโค้ดเพียงไม่กี่บรรทัด เพียงแค่ติดตั้ง Library 2 ตัว ที่มีชื่อว่า OneWire และ DallasTemperature สำหรับโค้ดที่ใช้แบบง่ายเราจะใช้คำสั่ง requestTemperatures() เพื่อเรียกข้อมูลจาก DS18B20 โดยข้อมูลนี้จะถูกส่งมาในรูปแบบสัญญาณดิจิทัล จึงต้องใช้คำสั่ง getTempCByIndex(ตำแหน่งเซ็นเซอร์) ของ Library เพื่อส่งค่าอุณหภูมิที่ได้ให้อยู่ในหน่วย องศาเซลเซียส (Fitrox Electronics, 2562)

6. เซ็นเซอร์วัดปริมาณสารที่ละลายในน้ำ TDS sensor



รูป 4 TDS Sensor

(ที่มา: arduinostep, ม.ป.ป.)

คุณสมบัติของ TDS Sensor ดังรูป 4 นั้นมีการทำงานที่กว้าง เพราะรองรับแรงดันไฟฟ้า 3.3-5.5V เป็นสัญญาณเอาต์พุตแบบอนาล็อกที่ 0-2.3V ซึ่งสามารถใช้กับระบบควบคุม 5V และ 3.3V ได้ เซ็นเซอร์ตัวนี้ใช้สัญญาณ AC เพื่อป้องกันการเกิดการกัดกร่อนของสัญญาณของเซ็นเซอร์ สามารถจุ่มในน้ำได้เป็นเวลานาน และใช้งานง่าย สามารถเชื่อมต่อกับ Arduino ได้โดยตรง ไม่ต้องบัดกรี (สุเทพ และ มาลัยวรรณ, 2561)

บริษัท นิโอนิคส์ (ม.ป.ป.) ได้ให้ความหมายไว้ว่า TDS ย่อมาจาก Total Dissolved Solids แปลเป็นภาษาไทยคือสารละลายทั้งหมดที่เหลือจากการระเหย โดยหมายถึงความเข้มข้นรวมทั้งหมดของสารที่ละลายในน้ำ ซึ่งอธิบายอย่างง่ายคือ ค่า TDS คือสารอะไรก็ตามไม่ว่าจะเป็นแร่ธาตุต่างๆ ที่ละลายเจือปนอยู่ในน้ำเช่นเกลือ สารอนินทรีย์และอินทรีย์วัตถุรวมถึงแร่ธาตุต่างๆ เช่นแคลเซียม แมกนีเซียม โพแทสเซียม และโซเดียมคาร์บอเนต ไนเตรต ไบคาร์บอเนต คลอไรด์ และซัลเฟต หรือ ระดับ TDS คือปริมาณของของแข็งที่ละลายในน้ำรวมทั้งหมด หน่วยวัดปริมาณ TDS คือ มก./ลิตร (mg/L) ซึ่งเท่ากับ ppm (parts per million) โดยความหมายคือน้ำหนักของแร่ธาตุ หรือสารต่างๆ ในหน่วยมิลลิกรัมในน้ำปริมาณ 1 ลิตร อีกทั้งยังนิยมใช้ กรัม/ลิตร (g/L) หรือ part per thousand (ppt) หรือ 1 ในพัน

วิธีวัดค่าการนำไฟฟ้าในน้ำ EC (Electrical Conductivity) อาศัยหลักการคือน้ำบริสุทธิ์ H_2O จะไม่นำไฟฟ้า แต่เมื่อน้ำมีการเจือปนด้วยแร่ธาตุหรือสารอื่นๆ ที่มีไอออนหรือประจุไฟฟ้าทำให้น้ำสามารถนำไฟฟ้าได้จากหลักการนี้และอาศัยค่าความสัมพันธ์ของ TDS และความนำไฟฟ้าจำเพาะของน้ำได้ดิน (Conductivity) สามารถประมาณได้โดยสมการต่อไปนี้ $TDS = k \cdot EC$ โดยที่ค่า TDS แสดงเป็นมิลลิกรัม/ลิตร และ EC คือค่าการนำไฟฟ้าในหน่วยไมโครซีเมนส์ต่อเซนติเมตรที่ 25 °C ค่า k คือปัจจัยการแปลงค่าจะแตกต่างกันระหว่าง 0.55 ถึง 0.8 เครื่องวัด TDS Meter จะใช้การวัดค่าการนำไฟฟ้านี้เพื่อแสดงค่า TDS โดยเครื่องวัดที่มีคุณภาพสามารถกำหนดปัจจัยการแปลงค่า (k) ได้ระหว่าง 0.1-1.0 ข้อเสียของวิธีนี้คือสารเคมีบางชนิดไม่เกิดไอออนทำให้ไม่สามารถวัดค่าการนำไฟฟ้าได้ แต่ข้อดีของวิธีนี้คือให้ค่าการวัดที่รวดเร็วทำได้โดยจุ่มโพรบของเครื่องวัดลงในน้ำที่ต้องการวัด ให้การวัดที่ความแม่นยำสูง

7. โมดูล Real Time Clock (RTC)



รูป 5 DS1302 RTC Module

(ที่มา: buraphatronics, 2562)

ร้าน arduitrionics (2557) ได้อธิบายโมดูล Real Time Clock (RTC) ดังรูป 5 ไว้ว่าเป็นอุปกรณ์ที่มีหน้าที่ในการให้ค่าเวลาตามจริง โดยการทำงานจะอาศัยสัญญาณนาฬิกาจาก Crystal oscillator เป็นแหล่งกำเนิดสัญญาณหลัก บางรุ่นของโมดูล RTC จะมาพร้อมกับแบตเตอรี่สำรองในตัว ซึ่งทำให้สามารถบันทึกเวลาอย่างต่อเนื่องได้แม้ในกรณีที่ไม่มีไฟเลี้ยงมาที่ตัวบอร์ด ฟังก์ชันนี้ช่วยลดความยุ่งยากในการตั้งค่าเวลาใหม่ทุกครั้งหลังจากที่อุปกรณ์ถูกปิดหรือไฟเลี้ยงขาด โมดูล RTC มีบทบาทสำคัญในงานที่ต้องการบันทึกข้อมูลพร้อมกับการประทับเวลา (Timestamp) เช่น การใช้งานในอุปกรณ์บันทึกข้อมูล (Data Logger) ที่ต้องการความถูกต้องและแม่นยำของข้อมูลเวลา

แม้ว่าไมโครคอนโทรลเลอร์ที่ใช้งานบน Arduino Board จะมีฟังก์ชันจับเวลา เช่น millis() ซึ่งสามารถใช้คำนวณเวลาที่ผ่านไปได้ แต่ฟังก์ชันนี้มีข้อจำกัดในแง่ของความแม่นยำ เนื่องจากไมโครโปรเซสเซอร์ที่เป็นหัวใจของ Arduino ต้องทำงานหลากหลายอย่างพร้อมกัน ตั้งแต่คำสั่งพื้นฐาน เช่น การคำนวณ ไปจนถึงการติดต่อกับอุปกรณ์ภายนอก เนื่องจากไมโครโปรเซสเซอร์ทำงานแบบอนุกรม (Serial) หรือทีละคำสั่ง ฟังก์ชันจับเวลาที่เขียนไว้ในโปรแกรมอาจถูกรบกวนจากการทำงานอื่น ๆ ที่แทรกเข้ามา (Interrupt) ส่งผลให้ค่าที่ได้จากฟังก์ชันเหล่านี้ไม่แม่นยำหรือไม่สามารถใช้งานได้ในกรณีที่ต้องการเวลาที่สอดคล้องกับความเป็นจริง

การแก้ปัญหาโดยใช้ RTC ในกรณีที่การประยุกต์ใช้งานต้องการเวลาที่แม่นยำและละเอียด เช่น วันที่ เดือน ปี ชั่วโมง นาที และวินาที การใช้งานโมดูล RTC จึงเป็นทางเลือกที่เหมาะสม โมดูลนี้ถูกออกแบบมาเฉพาะเพื่อทำหน้าที่ในการบันทึกเวลา และใช้สัญญาณนาฬิกาจาก Crystal oscillator แยกต่างหาก ทำให้สามารถทำงานได้อย่างอิสระและไม่ถูกรบกวนจากคำสั่งอื่นในระบบ

ปัจจุบันมีการออกแบบชิป RTC สำเร็จรูปที่ใช้งานง่าย เช่น DS1307 หรือ DS3231 โดยโมดูลเหล่านี้สามารถเชื่อมต่อกับไมโครคอนโทรลเลอร์ผ่านการสื่อสารแบบ I2C (Inter-Integrated Circuit) ซึ่งใช้สายสัญญาณเพียง 4 เส้น ได้แก่ SDA, SCL, VCC และ GND การออกแบบนี้ทำให้การใช้งาน RTC เป็นไปอย่างสะดวกและสามารถนำไปปรับใช้กับงานที่ต้องการความแม่นยำสูงได้อย่างง่ายดาย

8. การแจ้งเตือนผ่าน Telegram

การแจ้งเตือนผ่าน Telegram โดยใช้ ESP32 เป็นการนำความสามารถของไมโครคอนโทรลเลอร์ ESP32 มาผสานกับแพลตฟอร์มการสื่อสาร Telegram ซึ่งช่วยให้สามารถส่งข้อความแจ้งเตือนหรือข้อมูลจากเซ็นเซอร์ไปยังผู้ใช้งานได้แบบเรียลไทม์

Telegram เป็นแอปพลิเคชันส่งข้อความที่รองรับการสร้างบอทเพื่ออัตโนมัติในกระบวนการส่งข้อความ โดยกระบวนการนี้ประกอบด้วย 5 ขั้นตอนหลักดังนี้

1. การสร้างบอทใน Telegram

- เปิดแอป Telegram และค้นหา "BotFather" ซึ่งเป็นบอทที่ใช้สำหรับสร้างและจัดการบอทอื่น ๆ
- เริ่มการสนทนากับ BotFather และพิมพ์คำสั่ง /newbot เพื่อสร้างบอทใหม่
- ตั้งชื่อและกำหนดชื่อผู้ใช้ (username) ให้กับบอท โดยชื่อผู้ใช้ต้องลงท้ายด้วยคำว่า "bot" เช่น MyESP32Bot
- หลังจากสร้างบอทเสร็จสิ้น BotFather จะส่ง โทเค็น (Token) ซึ่งเป็นรหัสสำหรับเข้าถึงบอท

2. การรับ Chat ID ของผู้ใช้

- ค้นหาและเริ่มการสนทนากับบอทชื่อ "IDBot" ใน Telegram
- พิมพ์คำสั่ง /getid เพื่อรับ Chat ID ของคุณ ซึ่งจำเป็นสำหรับการส่งข้อความไปยังผู้ใช้

3. การตั้งค่า ESP32

- ติดตั้งไลบรารี Universal-Arduino-Telegram-Bot ใน Arduino IDE โดยสามารถดาวน์โหลดได้จาก GitHub
- ตั้งค่า Wi-Fi SSID และรหัสผ่าน รวมถึงระบุโทเค็นและ Chat ID ที่ได้รับในขั้นตอนก่อนหน้านี้ในโค้ดของ ESP32

4. การเขียนโปรแกรมเพื่อส่งข้อความ

ใช้ฟังก์ชันจากไลบรารี Universal-Arduino-Telegram-Bot เพื่อเชื่อมต่อ ESP32 กับเซิร์ฟเวอร์ Telegram และส่งข้อความ ตัวอย่างโค้ดมีดังนี้:

```
#include <WiFi.h>
#include <UniversalTelegramBot.h>
#include <WiFiClientSecure.h>
const char* ssid = "Your_SSID";
const char* password = "Your_PASSWORD";
#define BOT_TOKEN "Your_Bot_Token"
#define CHAT_ID "Your_Chat_ID"
WiFiClientSecure client;
UniversalTelegramBot bot(BOT_TOKEN, client);
void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("WiFi connected!");
}
void loop() {
  bot.sendMessage(CHAT_ID, "Hello from ESP32!", "");
  delay(10000); // ส่งข้อความทุก 10 วินาที
}
```

5. การติดตั้งและอัปโหลดโค้ด

- เชื่อมต่อ ESP32 กับคอมพิวเตอร์ผ่านสาย USB
- เปิด Arduino IDE และเขียนโค้ดตามตัวอย่าง
- ติดตั้งไลบรารี Universal-Arduino-Telegram-Bot หากยังไม่ได้ติดตั้ง
- อัปโหลดโค้ดไปยัง ESP32
- ตรวจสอบว่า ESP32 เชื่อมต่อกับ Wi-Fi และสามารถส่งข้อความไปยัง Telegram ได้สำเร็จ

(ElectronicWings, ม.ป.ป.)

9. การส่งข้อมูลจาก ESP32 ไปยัง Google Sheets

การเชื่อมต่อ ESP32 กับ Google Sheets เป็นหนึ่งในวิธีที่มีประสิทธิภาพสำหรับการจัดเก็บและบริหารจัดการข้อมูลในโครงการ IoT โดย Google Sheets ช่วยให้สามารถเก็บข้อมูลแบบเรียลไทม์ ทั้งยังรองรับการวิเคราะห์และการแสดงผลข้อมูลในรูปแบบกราฟหรือรายงานได้อย่างง่ายดาย

ในโครงการนี้ เลือกใช้วิธีการเชื่อมต่อที่อ้างอิงจาก ElectroPeak โดยแบ่งขั้นตอนการดำเนินงานออกเป็น 3 ส่วนหลัก ดังนี้

1. การตั้งค่า Google Sheets

- สร้าง Google Sheets ใหม่
- ใช้ Google Apps Script ในการสร้าง Web App ที่สามารถรับข้อมูลจาก ESP32 ผ่าน HTTP POST
- ตัวอย่าง Google Apps Script ที่ใช้ในการรับข้อมูล:

```
function doPost(e) {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var data = JSON.parse(e.postData.contents);
  sheet.appendRow([data.time, data.temperature]);
  return ContentService.createTextOutput("Success");
}
```

สคริปต์นี้จะเพิ่มข้อมูลที่รับจาก ESP32 ลงในแถวใหม่ของ Google Sheets โดยข้อมูลประกอบด้วย เวลา (time) และอุณหภูมิ (temperature)

2. การเขียนโปรแกรม ESP32

- ใช้ไลบรารี WiFi.h และ HTTPClient.h เพื่อให้ ESP32 สามารถเชื่อมต่อ Wi-Fi และส่งข้อมูลผ่าน HTTP POST ไปยัง Web App

- ตัวอย่างโค้ด ESP32:

```
#include <WiFi.h>
#include <HTTPClient.h>

const char* ssid = "Your_SSID";    // ชื่อ Wi-Fi
const char* password = "Your_PASSWORD"; // รหัสผ่าน Wi-Fi
const char* scriptUrl = "https://script.google.com/macros/s/Your_Script_ID/exec"; //
URL ของ Web App

void setup() {
  WiFi.begin(ssid, password); // เชื่อมต่อ Wi-Fi
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
  }
}

void loop() {
  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;
    http.begin(scriptUrl);
    http.addHeader("Content-Type", "application/json");
    // ข้อมูลตัวอย่าง
    String jsonData = "{\"time\":\"2025-01-14T10:00:00Z\",\"temperature\":25.5}";
    int httpResponseCode = http.POST(jsonData);
    http.end();
  }
  delay(60000); // ส่งข้อมูลทุก 1 นาที
}
```

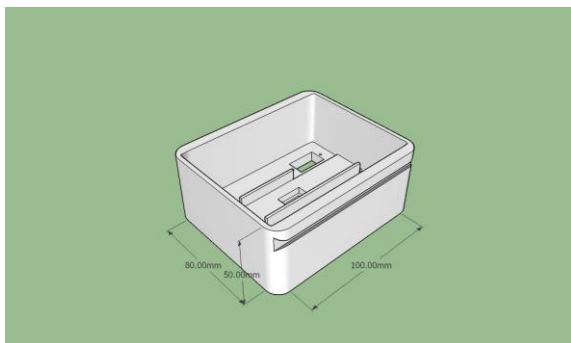
3. การทดสอบ

- ตรวจสอบว่า ESP32 เชื่อมต่อกับ Wi-Fi ได้สำเร็จ
- ทดสอบการส่งข้อมูล โดยตรวจสอบข้อมูลที่แสดงใน Google Sheets ว่าถูกต้องหรือไม่

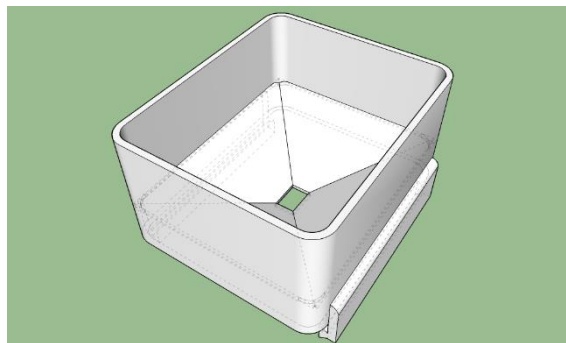
(ElectroPeak, ม.ป.ป.)

10. เครื่องให้อาหาร

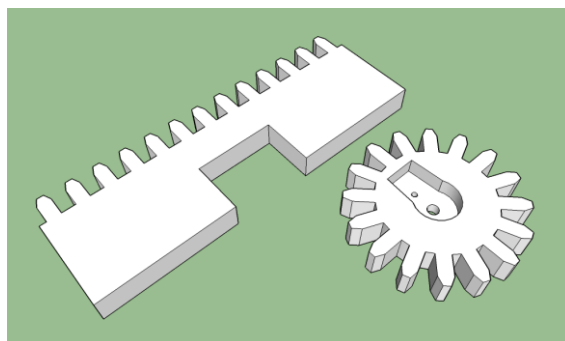
ในโครงงานนี้ เครื่องให้อาหารถูกออกแบบและผลิตโดยผู้จัดทำเองดังรูป 6, 7 และ 8 โดยใช้ซอฟต์แวร์ Sketchup ที่ใช้สำหรับสร้างโมเดล 3 มิติ ซึ่งเป็นซอฟต์แวร์ที่ใช้งานง่ายและเหมาะสำหรับงานออกแบบเชิงสร้างสรรค์ จากนั้นชิ้นงานบางชิ้นที่ใช้เวลาในการพิมพ์ที่เหมาะสม เช่น เฟือง ดังรูป 8 จะถูกพิมพ์ด้วยเครื่องพิมพ์ 3 มิติ โดยใช้วัสดุ PLA (Polylactic Acid) ซึ่งมีคุณสมบัติแข็งแรง น้ำหนักเบา และเป็นมิตรต่อสิ่งแวดล้อม และชิ้นงานที่ใช้เวลาพิมพ์มาก ผู้จัดทำจะตัดแผ่นพลาสติก (Plaswood) และประกอบตามแบบที่ออกแบบไว้



รูป 6 ฐานเครื่องให้อาหาร



รูป 7 ที่ใส่อาหาร



รูป 8 เฟืองปิด-เปิดช่องใส่อาหาร

หลักการทำงานคือ เครื่องให้อาหารนี้ใช้ เฟืองราง ที่ถูกออกแบบให้สามารถเลื่อนเพื่อปิด-เปิดการปล่อยอาหาร โดยเฟืองรางจะถูกขับเคลื่อนด้วย เฟืองตรง ที่ยึดติดกับมอเตอร์เซอร์โว (Servo Motor) เพื่อหมุนและควบคุมการเปิด-ปิดของช่องให้อาหาร ช่องใส่อาหารออกแบบให้มีลักษณะเป็นรูปกรวยสี่เหลี่ยมที่ช่วยให้อาหารไหลลงได้ง่าย โดยมีช่องตรงกลางสำหรับปล่อยอาหารในปริมาณที่กำหนด

โดยระบบควบคุมนั้น จะใช้มอเตอร์เซอร์โวที่ถูกควบคุมโดยไมโครคอนโทรลเลอร์ผ่านสัญญาณ PWM (Pulse Width Modulation) ซึ่งช่วยปรับมุมการหมุนของมอเตอร์เซอร์โวอย่างแม่นยำ ทำให้สามารถควบคุมการปล่อยอาหารได้ตามเวลาที่ตั้งไว้หรือปริมาณที่ต้องการได้อย่างมีประสิทธิภาพ

อย่างไรก็ตาม เนื่องจากไมโครคอนโทรลเลอร์ ESP32 ที่ใช้ในโครงงานทำงานที่แรงดันไฟฟ้าระดับ 3.3V ขณะที่มอเตอร์เซอร์โวที่ใช้อาจรับแรงดันสัญญาณควบคุมระดับ 5V จึงจำเป็นต้องใช้ Level Shifter (ตัวปรับระดับแรงดันไฟฟ้า) เพื่อปรับสัญญาณ PWM จาก 3.3V ของไมโครคอนโทรลเลอร์ให้เป็น 5V ที่เหมาะสมกับมอเตอร์เซอร์โว การใช้ Level Shifter ช่วยให้อุปกรณ์ทำงานได้อย่างถูกต้องและเสถียร พร้อมทั้งป้องกันปัญหาที่อาจเกิดจากแรงดันสัญญาณที่ไม่สอดคล้องกัน

11. อุปกรณ์ควบคุมสภาพแวดล้อม

อุปกรณ์ต่าง ๆ ที่นำมาใช้ในโครงการนี้เป็นอุปกรณ์ที่มีหน้าที่ควบคุมสภาพแวดล้อมของน้ำในบ่อเลี้ยงกุ้งฝอย เพื่อให้เหมาะสมกับการเจริญเติบโต โดยในโครงการนี้ใช้อุปกรณ์ควบคุมสภาพแวดล้อม 4 ประเภท ได้แก่ เครื่องให้ออกซิเจน เครื่องกรองน้ำ เครื่องระบายความร้อน และเครื่องทำความร้อน ดังนี้

1. เครื่องให้ออกซิเจน



รูป 9 เครื่องให้ออกซิเจน

(ที่มา: TARAD, ม.ป.ป.)

เครื่องให้ออกซิเจน (Air Pump) ดังรูป 9 มีหน้าที่เพิ่มออกซิเจนในน้ำโดยดึงอากาศจากภายนอกเข้าสู่ตู้เลี้ยงกุ้งฝอยผ่านหัวทราย รุ่นที่ใช้ในโครงการนี้เป็นรุ่น SUNSUN QB-101B ที่มีความสามารถในการผลิตแรงลมได้สูงถึง 18.6 +/- 10% ลิตร/ชั่วโมง ทำงานเงียบ น้อยกว่า 33 เดซิเบล และใช้พลังงานแค่ 1.5 วัตต์ (TARAD, ม.ป.ป.)

2. เครื่องกรองน้ำ



รูป 10 เครื่องกรองน้ำ

(ที่มา: TARAD, ม.ป.ป.)

เครื่องกรองน้ำในระบบเลี้ยงกุ้งฝอยดังรูป 10 มีบทบาทสำคัญในการกำจัดสิ่งสกปรกในน้ำ รวมถึงสารเคมีที่อาจเป็นอันตรายต่อกุ้งฝอย รุ่นที่ใช้ในโครงการนี้เป็นรุ่น Jeneca XP-06 ที่มีความสามารถในการกรองน้ำได้ 250 ลิตร/ชั่วโมง และใช้พลังงานแค่ 3.5 วัตต์ (TARAD, ม.ป.ป.)

3. เครื่องระบายความร้อน



รูป 11 คอยล์เย็น

(ที่มา: amazon, ม.ป.ป.)

เครื่องระบายความร้อนดังรูป 11 ในโครงการนี้ใช้หลักการนำความร้อนจากน้ำผ่าน คอยล์เย็น (Cooling Coil) และระบายออกด้วยพัดลมขนาด 120 มม. ที่ถูกสูบน้ำขึ้นมาผ่านปั้มน้ำ DC 12 โวลต์ และระบบนี้ถูกออกแบบคล้ายกับระบบระบายความร้อนในคอมพิวเตอร์ ช่วยรักษาอุณหภูมิของน้ำให้อยู่ในช่วงที่เหมาะสมต่อการเลี้ยงกุ้งฝอย

4. เครื่องทำความร้อน



รูป 12 เครื่องทำความร้อน

(ที่มา: smile petlism shop, ม.ป.ป.)

เครื่องทำความร้อน (Heater) ใช้เพื่อเพิ่มอุณหภูมิของน้ำในกรณีที่อุณหภูมิต่ำกว่าช่วงที่เหมาะสม รุ่นที่ใช้ในโครงการนี้คือ SUNSUN GR-50B ดังรูป 12 ที่มีความสามารถในการปรับอุณหภูมิอัตโนมัติ และใช้พลังงาน 50 วัตต์

บทที่ 3

อุปกรณ์และวิธีการพัฒนานวัตกรรม

อุปกรณ์และเครื่องมือที่ใช้การออกแบบการสร้าง

1. วัสดุอุปกรณ์

1. แผ่นไม้อัด
2. สายไฟ
3. สว่าน
4. สกรู
5. คีมตัด
6. ไชควงหัวแฉก
7. ปืนกาว
8. กาวแท่ง
9. หัวแร้ง
10. เครื่องพิมพ์ 3 มิติ
11. สายยางใส 2 หุน หรือ ¼ นิ้ว
12. แผ่นพลาสติก
13. แผ่นอะคริลิกใส
14. บานพับประตู
15. แผ่นพลาสติกปูบ่อปลาสีดำ
16. ตู้ปลาขนาด 20 x 10 x 10 นิ้ว
17. ปุ๋ยมูลไก่
18. สหรัยทางกระรอก
19. กรวดและทราย

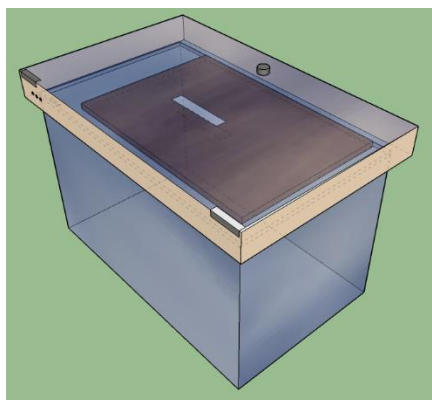
2. อุปกรณ์อิเล็กทรอนิกส์

1. ไมโครคอนโทรลเลอร์ ESP32 DEVKIT V1
2. RTC DS1302 Real Time Clock Module
3. จอ OLED SSD1309 2.42 นิ้ว 4Pin I2C
4. NA019 โมดูลแปลงระดับสัญญาณ logic 4 ช่อง
5. เซ็นเซอร์วัดอุณหภูมิ DS18B20
6. เซ็นเซอร์วัดปริมาณสารที่ละลายในน้ำ TDS sensor
7. ฮีตเตอร์ทำความร้อน SUNSUN GR-50B
8. ปั๊มลม QB-101B
9. เครื่องกรองน้ำ Jeneca XP-06
10. ปั๊มน้ำ DC 12v
11. คอยล์เย็นระบายความร้อน 120mm x 120mm
12. พัดลม DC 12v 120mm x 120mm
13. มอเตอร์ Servo SG90 180 องศา
14. รีเลย์ 5v
15. แผ่นวงจร PCB
16. อะแดปเตอร์ 12v 5A
17. โมดูลแปลงไฟ Step Down
18. เบรกเกอร์ 10A

วิธีการดำเนินการ

1. ศึกษาการสร้างระบบเลี้ยงกุ้งฝอยโดยใช้เทคโนโลยี IoT

- 1.1. ศึกษาข้อมูลเกี่ยวกับการเลี้ยงกุ้งฝอย ความต้องการของระบบ IoT และ ออกแบบภาพร่างระบบ โดยรวมดังรูป 13 เช่น การทำงานของวงจร การติดตั้งเซนเซอร์ และตำแหน่งเครื่องให้อาหาร



รูป 13 ออกแบบระบบเลี้ยงกุ้งฝอย

1.2. การออกแบบและพัฒนาเครื่องให้อาหารอัตโนมัติ

1. ออกแบบเครื่องให้อาหารโดยใช้ซอฟต์แวร์ Sketchup ที่ใช้สำหรับสร้างโมเดล 3 มิติ
2. พิมพ์ชิ้นส่วนพิมพ์ด้วยเครื่องพิมพ์ 3 มิติ
3. ตัดแผ่นพลาสติกให้ได้ขนาดสำหรับใช้เป็นฐานและตัวโครงเครื่องให้อาหาร โดยเครื่องให้อาหารที่ออกแบบไว้จะมีขนาด 120mm x 100mm
4. ประกอบชิ้นส่วนต่าง ๆ ของเครื่องให้อาหาร เช่น ฟันเฟือง มอเตอร์เซอร์โว และที่ใส่อาหาร

1.3. การติดตั้งระบบ IoT

1. เขียนโค้ดสำหรับควบคุมระบบโดยใช้ภาษา C++ ในโปรแกรม Arduino IDE
2. ตั้งค่าการเชื่อมต่อ IoT ผ่าน WiFi เพื่อเชื่อมต่อกับ Google Sheets และ Telegram
3. ทดสอบระบบการส่งข้อมูลจากเซนเซอร์ไปยังแพลตฟอร์ม IoT

1.4. ติดตั้งและประกอบระบบในตู้เลี้ยงกุ้งฝอย

1. ตัดแผ่นพลาสติกให้พอดีกับขนาดตู้ และเจาะรูสำหรับติดตั้งเซนเซอร์วัดอุณหภูมิ เซนเซอร์วัดค่า TDS และอุปกรณ์ควบคุมสภาพแวดล้อม
2. ติดตั้งแผงวงจรบนแผ่นไม้อัด แล้วนำมาติดตั้งเข้ากับพลาสติกด้วยกาวแท่ง
3. ติดตั้งเครื่องกรองน้ำ เครื่องให้ออกซิเจน เครื่องทำความร้อน เครื่องระบายความร้อนให้ยึดติดกับแผ่นไม้อัด
4. ติดตั้งแผ่นอะคริลิกใสพร้อมบานพับ เพื่อใช้เป็นประตูสำหรับเปิดตรวจสอบวงจร
5. ติดตั้งเครื่องให้อาหารที่ประกอบสำเร็จเข้ากับตัวฝาของตู้ดังรูป 14



รูป 14 ระบบเลี้ยงกุ้งฝอย IoT

1.5. การทดสอบระบบและปรับปรุง

1. ทดสอบการทำงานของเครื่องให้อาหาร อุปกรณ์ควบคุมสภาพแวดล้อม เซนเซอร์วัดอุณหภูมิ และเซนเซอร์วัด TDS
2. ตรวจสอบระบบ IoT เช่น การแจ้งเตือนเมื่อค่าคุณภาพน้ำเปลี่ยน
3. แก้ไขปัญหาที่พบในระบบ และปรับปรุงให้การทำงานเสถียรมากที่สุด

1.6. ประเมินประสิทธิภาพของระบบ อัตราการรอดของกุ้งฝอย และการแจ้งเตือนของระบบ

2. ทดสอบประสิทธิภาพการทำงานของระบบเลี้ยงกุ้งฝอยโดยใช้เทคโนโลยี IoT

2.1. ทดสอบประสิทธิภาพของระบบเลี้ยงกุ้งฝอย IoT ด้วยการเตรียมบ่อที่มีปุ๋ยมูลไก่ กรวดและทราย

2.1.1. การเตรียมเลี้ยงกุ้งฝอย

1. นำน้ำเปล่าปริมาตร 10 ลิตรมาใส่ถังใส่น้ำ และใส่ปุ๋ยมูลไก่ในอัตราส่วน 120 กิโลกรัมต่อไร่ หรือ 9.6 กรัมต่อขนาดตู้ 10 x 20 ตารางนิ้ว และทิ้งน้ำไว้ 4 วันจนมูลไก่ผสมเข้าไปกับน้ำ
2. นำสาหร่ายหางกระรอกมาไว้ในน้ำ 15 ต้น
3. ใส่กรวด 1.5 กิโลกรัมและทราย 1 กิโลกรัมเข้าไปในตู้
4. นำน้ำที่ผสมปุ๋ยมูลไก่และใส่สาหร่ายหางกระรอกแล้ว นำมาใส่ในตู้กระจกและเติมน้ำเพิ่มอีก 10 ลิตร

2.1.2. นำกุ้งฝอยมาเลี้ยงในตู้กระจกที่เตรียมไว้ 20 ตัว

2.1.3. นำฟัดูที่มีระบบควบคุมมาปิดตู้

2.1.4. เปิดระบบควบคุมเลี้ยงกุ้งฝอยไว้เป็นเวลา 3 วัน และทำการวัดอัตราการรอด

2.2. ทดสอบประสิทธิภาพของระบบเลี้ยงกุ้งฝอย IoT ด้วยการเตรียมบ่อที่ไม่มีปุ๋ยมูลไก่ กรวดและทราย

2.2.1. การเตรียมเลี้ยงกุ้งฝอย

1. นำน้ำใส่ในตู้กระจก 20 ลิตร
2. นำสาหร่ายหางกระรอกมาไว้ในน้ำ 15 ต้น

2.2.2. นำกุ้งฝอยมาเลี้ยงในตู้กระจกที่เตรียมไว้ 20 ตัว

2.2.3. นำฟัดูที่มีระบบควบคุมมาปิดตู้

2.2.4. เปิดระบบควบคุมเลี้ยงกุ้งฝอยไว้เป็นเวลา 3 วัน และทำการวัดอัตราการรอด

บทที่ 4

ผลการดำเนินงาน

การสร้างระบบเลี้ยงกุ้งฝอย IOT ให้มีประสิทธิภาพในการเลี้ยงกุ้งฝอย คณะผู้จัดทำโครงการได้ดำเนินการตามขั้นตอนที่ได้วางแผนไว้

ตอนที่ 1 ศึกษาการสร้างระบบเลี้ยงกุ้งฝอยโดยใช้เทคโนโลยี IoT

จากการศึกษาหลักการและวิธีในการสร้างระบบเลี้ยงกุ้งฝอย IoT ซึ่งอาศัยหลักการการควบคุมสภาพแวดล้อมให้เหมาะสม ได้ระบบเลี้ยงกุ้งฝอยโดยใช้เทคโนโลยี IoT ที่วัดอุณหภูมิและค่า TDS เพื่อควบคุมการทำงานของระบบเลี้ยงกุ้งฝอยด้วยเครื่องกรองน้ำ เครื่องให้ออกซิเจน เครื่องทำความร้อน และเครื่องระบายความร้อน

ตอนที่ 2 ทดสอบประสิทธิภาพการทำงานของระบบเลี้ยงกุ้งฝอยโดยใช้เทคโนโลยี IoT

2.1. ทดสอบประสิทธิภาพของระบบเลี้ยงกุ้งฝอย IoT ด้วยการเตรียมบ่อที่มีปุ๋ยมูลไก่ กรวดและทราย

จากการศึกษาการเลี้ยงกุ้งฝอยโดยใช้ระบบเลี้ยงกุ้งฝอยแบบ IoT ด้วยการเตรียมบ่อที่มีปุ๋ยมูลไก่ ใส่กรวดและทรายในบ่อดังรูป 15 พบว่าอัตราการรอดของกุ้งฝอยเป็นร้อยละ 25 ของกุ้งฝอยทั้งหมด โดยอุณหภูมิจะอยู่ในช่วง 26 – 27 องศาเซลเซียส ซึ่งเป็นค่าที่เหมาะสมกับกุ้งฝอยโดยสอดคล้องกับผลการทดลองของ ศุภวิชญ์ ขวลิทธิเชษฐ (2561) และปริมาณของแข็งที่ละลายในน้ำ (TDS) อยู่ในช่วง 60 – 70 PPM ดังตาราง 1 ซึ่งต่ำกว่าค่าที่เหมาะสมกับกุ้งคือ 80 – 150 PPM ซึ่งมาจากผลการทดลองของ อติราช สุขสวัสดิ์ และคณะ (2566)

ตาราง 1 ประสิทธิภาพของระบบเลี้ยงกุ้งฝอย IOT ด้วยการเตรียมบ่อที่มีปุ๋ยมูลไก่ กรวดและทราย

หัวข้อ	ผลการทดลอง
อัตราการรอดของกุ้งฝอย	25%
อุณหภูมิในบ่อ	26 – 27 องศาเซลเซียส
ปริมาณของแข็งที่ละลายในน้ำ (TDS)	60 – 70 PPM

จากผลการทดลองชี้ให้เห็นว่าบ่อที่มีปุ๋ยมูลไก่ กรวด และทรายสามารถรักษาอุณหภูมิและค่า TDS ให้คงที่ได้ แต่ยังคงมีอัตราการรอดของกุ้งฝอยต่ำ ซึ่งสันนิษฐานว่าเกิดจากผลกระทบของปุ๋ยมูลไก่ทำให้จำนวนออกซิเจนไม่เพียงพอสำหรับการเลี้ยงกุ้ง ซึ่งสอดคล้องกับงานวิจัยของ ขจรเกียรติ ศรีนวลสม (2553) ทรายและกรวดทำให้เกิดการหมักหมมของสารอินทรีย์ ทำให้เกิดการเน่าเสียของน้ำเนื่องจากอาหารที่ตกค้างในบ่อ



รูป 15 ทดสอบประสิทธิภาพด้วยการเตรียมน้ำที่มีปุ๋ยมูลไก่ กรวดและทราย

2.2. ทดสอบประสิทธิภาพของระบบเลี้ยงกุ้งฝอย IoT ด้วยการเตรียมน้ำที่ไม่มีปุ๋ยมูลไก่ กรวดและทราย

จากการศึกษาการเลี้ยงกุ้งฝอยโดยใช้ระบบเลี้ยงกุ้งฝอยแบบ IoT ด้วยการเตรียมน้ำที่ไม่มีปุ๋ยมูลไก่ ไม่ใส่กรวดและทรายในบ่อดังรูป 16 พบว่าอัตราการรอดของกุ้งฝอยเป็นร้อยละ 100 ของกุ้งฝอยทั้งหมด โดยอุณหภูมิจะอยู่ในช่วง 26 – 27 องศาเซลเซียส และปริมาณของแข็งที่ละลายในน้ำ (TDS) อยู่ที่ 0 PPM และจะเพิ่มขึ้นไปเรื่อยๆจนถึง 30 PPM ดังตารางที่ 2

ตาราง 2 ประสิทธิภาพของระบบเลี้ยงกุ้งฝอย IoT ด้วยการเตรียมน้ำที่ไม่มีปุ๋ยมูลไก่ กรวดและทราย

หัวข้อ	ผลการทดลอง
อัตราการรอดของกุ้งฝอย	100%
อุณหภูมิในบ่อ	26 – 27 องศาเซลเซียส
ปริมาณของแข็งที่ละลายในน้ำ (TDS)	0 – 30 PPM

ผลการทดลองแสดงให้เห็นว่าสภาพแวดล้อมที่ไม่มีปุ๋ยมูลไก่ กรวดและทราย สามารถรักษาอุณหภูมิและค่า TDS ให้คงที่ได้ และช่วยลดปัจจัยเสี่ยงต่อการเลี้ยงกุ้งฝอย ทำให้อัตราการรอดสูงขึ้น



รูป 16 ทดสอบประสิทธิภาพด้วยการเตรียมน้ำที่ไม่มีปุ๋ยมูลไก่ กรวดและทราย

บทที่ 5

สรุปผลการดำเนินการ/อภิปรายผลการดำเนินการ

สรุปผลการดำเนินงาน

ตอนที่ 1 ศึกษาการสร้างระบบเลี้ยงกุ้งฝอยโดยใช้เทคโนโลยี IoT

จากการศึกษาหลักการและวิธีสร้างระบบเลี้ยงกุ้งฝอย IoT ซึ่งอาศัยหลักการการควบคุมสภาพแวดล้อมให้เหมาะสม ได้ระบบเลี้ยงกุ้งฝอยโดยใช้เทคโนโลยี IoT ที่วัดอุณหภูมิและค่า TDS เพื่อควบคุมการทำงานของระบบเลี้ยงกุ้งฝอยด้วยเครื่องกรองน้ำ เครื่องให้ออกซิเจน เครื่องทำความร้อน และเครื่องระบายความร้อน

ตอนที่ 2 ทดสอบประสิทธิภาพการทำงานของระบบเลี้ยงกุ้งฝอยโดยใช้เทคโนโลยี IoT

1. ทดสอบประสิทธิภาพของระบบเลี้ยงกุ้งฝอย IoT ด้วยการเตรียมบ่อที่มีปุ๋ยมูลไก่กวดและทราย

จากผลการทดลองชี้ให้เห็นว่าบ่อที่มีปุ๋ยมูลไก่ กวด และทรายสามารถรักษาอุณหภูมิและค่า TDS ให้คงที่ได้ แต่ยังคงมีอัตราการรอดของกุ้งฝอยต่ำ

2. ทดสอบประสิทธิภาพของระบบเลี้ยงกุ้งฝอย IoT ด้วยการเตรียมบ่อที่ไม่มีปุ๋ยมูลไก่กวดและทราย

ผลการทดลองแสดงให้เห็นว่าสภาพแวดล้อมที่ไม่มีปุ๋ยมูลไก่ กวดและทราย สามารถรักษาสภาพแวดล้อมของอุณหภูมิและค่า TDS ให้คงที่ได้ และมีอัตราการรอดที่เพิ่มขึ้น

อภิปรายผลการทดลอง

ตอนที่ 1 ศึกษาการสร้างระบบเลี้ยงกุ้งฝอยโดยใช้เทคโนโลยี IoT

จากการศึกษาหลักการและวิธีการสร้างระบบเลี้ยงกุ้งฝอยโดยใช้เทคโนโลยี IoT พบว่าระบบสามารถควบคุมสภาพแวดล้อมให้เหมาะสมต่อการเลี้ยงกุ้งฝอยได้สำเร็จ ทั้งการควบคุมอุณหภูมิและค่าของแข็งที่ละลายในน้ำ (TDS) ได้คงที่

ตอนที่ 2 ทดสอบประสิทธิภาพการทำงานของระบบเลี้ยงกุ้งฝอยโดยใช้เทคโนโลยี IoT

การทดลองในบ่อที่มีปุ๋ยมูลไก่ กวด และทราย พบว่าอัตราการรอดของกุ้งฝอยอยู่ที่ 25% แม้ว่าอุณหภูมิจะเหมาะสมและ TDS ใกล้เคียงค่าที่เหมาะสม แต่สันนิษฐานว่าเกิดจากผลกระทบของปุ๋ยมูลไก่ทำให้จำนวนออกซิเจนไม่เพียงพอสำหรับการเลี้ยงกุ้ง ซึ่งสอดคล้องกับงานวิจัยของ ขจรเกียรติ ศรีนวลสม (2553) ทรายและกวดทำให้เกิดการหมักหมมของสารอินทรีย์ ทำให้เกิดการเน่าเสียของน้ำเนื่องจากอาหารที่ตกค้างในบ่อ ที่อาจส่งผลกระทบต่ออัตราการรอด และในบ่อที่ไม่มีปุ๋ยมูลไก่ กวด และทราย พบว่าอัตราการรอดของกุ้งฝอยอยู่ที่ 100% ถึงแม้ว่าค่า TDS จะต่ำกว่าค่าที่เหมาะสม แต่สภาพแวดล้อมที่ปราศจากสารอินทรีย์ที่หมักหมมช่วยลดปัจจัยที่เป็นอันตรายต่อการเลี้ยง ทำให้อัตราการรอดสูงขึ้น

ข้อเสนอแนะ

คณะผู้จัดทำโครงการมีข้อเสนอแนะสำหรับผู้ที่ต้องการพัฒนาระบบเลี้ยงกุ้งฝอย IoT ได้ดังนี้

1. ควรวัดการเจริญเติบโต เพื่อให้ทราบคุณภาพของกุ้งฝอยได้ชัดเจนมากยิ่งขึ้น
2. ควรใช้เครื่องให้ออกซิเจนที่มีคุณภาพมากขึ้น เพื่อให้ประสิทธิภาพในการเลี้ยงกุ้งฝอยดียิ่งขึ้น
3. ควรมีการวัดปริมาณออกซิเจนที่ละลายในน้ำ เพื่อให้มีข้อมูลในการอภิปรายผลได้มากและชัดเจนขึ้น
4. การเชื่อมต่อวงจร การใช้อุปกรณ์ ควรมีความรอบครอบ และมีผู้เชี่ยวชาญคอยให้คำปรึกษา

เอกสารอ้างอิง

- Arduitrronics. (2557). **Real Time Clock DS3231**. [ออนไลน์]. เข้าถึงได้จาก: <https://www.arduitronics.com/>. (วันที่สืบค้นข้อมูล: 1 ธันวาคม 2567).
- ElectronicWings. (ม.ป.ป.). **Send a Telegram message using ESP32**. [ออนไลน์]. เข้าถึงได้จาก: <https://www.electronicwings.com/>. (วันที่สืบค้นข้อมูล: 25 มกราคม 2568).
- Fitrox Electronics. (2562). **การใช้งาน DS18B20 Digital Temperature**. [ออนไลน์]. เข้าถึงได้จาก: <http://fitrox.lnwshop.com/>. (วันที่สืบค้นข้อมูล: 1 ธันวาคม 2567).
- GROBOTRONICS. (ม.ป.ป.). **ESP32 Development Board - DEVKIT V1**. [ออนไลน์]. เข้าถึงได้จาก: <https://grobotronics.com/>. (วันที่สืบค้นข้อมูล: 1 ธันวาคม 2567).
- IOT.CodeMobiles. (ม.ป.ป.). **เซนเซอร์วัดอุณหภูมิในน้ำ DS18B20 สายยาว 1 เมตร Digital Temperature Sensor Probe Waterproof**. [ออนไลน์]. เข้าถึงได้จาก <http://www.iot.codemobiles.com/>. (วันที่สืบค้นข้อมูล: 1 ธันวาคม 2567).
- Plantlover. (ม.ป.ป.). **สาหร่ายทางกระจก รอก ปลุกในตู้ปลา เพิ่มออกซิเจน ประโยชน์เยอะ!!**. [ออนไลน์]. เข้าถึงได้จาก: <https://plantlover.net/>. (วันที่สืบค้นข้อมูล: 25 มกราคม 2568).
- TARAD. (ม.ป.ป.). **Jeneca XP-06**. [ออนไลน์]. เข้าถึงได้จาก: <https://thaiaquariumcenter.tarad.com/>. (วันที่สืบค้นข้อมูล: 1 ธันวาคม 2567).
- TARAD. (ม.ป.ป.). **SUNSUN Air Pump QB-101B ปั๊มออกซิเจน 1 ทาง ขนาดเล็ก เสียงเงียบมาก แบบจุกยาง Small for Hang on**. [ออนไลน์]. เข้าถึงได้จาก: <https://thaiaquariumcenter.tarad.com/>. (วันที่สืบค้นข้อมูล: 1 ธันวาคม 2567).
- amazon. (ม.ป.ป.) **G 1/4 18 Tubes Radiator Water Cooler Heat Exchanger Aluminium Heating System 120 mm**. [ออนไลน์]. เข้าถึงได้จาก: <https://www.amazon.de/>. (วันที่สืบค้นข้อมูล: 1 ธันวาคม 2567).
- arduinostep. (ม.ป.ป.). **TDS Meter เซนเซอร์วัดคุณภาพน้ำ**. [ออนไลน์]. เข้าถึงได้จาก: <http://www.arduinostep.com/>. (วันที่สืบค้นข้อมูล: 1 ธันวาคม 2567).
- buraphatronics (2562). **DS1302 RTC Module**. [ออนไลน์]. เข้าถึงได้จาก: <https://www.buraphatronics.com/>. (วันที่สืบค้นข้อมูล: 1 ธันวาคม 2567).
- gosmartfarmer. (2567). **ม.แม่โจ้แนะวิธีเลี้ยงกุ้งฝอย อัตรารอด 80% เลี้ยงง่าย ใช้เวลา 2 เดือนขายได้**. [ออนไลน์]. เข้าถึงได้จาก: <https://www.gosmartfarmer.com/>. (วันที่สืบค้นข้อมูล: 1 ธันวาคม 2567).
- smile petlism shop. (ม.ป.ป.). **ฮีตเตอร์ตู้ปลา Heater SUNSUN รุ่น GR 50/100/200/300/500w สวม ปลอกกันกระแทกอย่างดี**. [ออนไลน์]. เข้าถึงได้จาก: <https://shopee.co.th/>. (วันที่สืบค้นข้อมูล: 1 ธันวาคม 2567).

- ขจรเกียรติ ศรีนวลสม. (2553). ผลของช่วงระยะเวลาที่ใช้ปุ๋ยมูลไก่ต่อการเปลี่ยนแปลงองค์ประกอบชนิดของแพลงก์ตอนพืชและผลผลิตปลานิลในบ่อเพาะเลี้ยงเชิงพาณิชย์. [ออนไลน์]. เข้าถึงได้จาก: <https://erp.mju.ac.th/>. (วันที่สืบค้นข้อมูล: 31 มกราคม 2568).
- นีโอนิคส์. (ม.ป.ป.). TDS คือ. [ออนไลน์]. เข้าถึงได้จาก: <https://www.tools.in.th/>. (วันที่สืบค้นข้อมูล: 1 ธันวาคม 2567).
- บัญชา ทองมี. (2555). การเพาะเลี้ยงกุ้งฝอยเชิงพาณิชย์. [ออนไลน์]. เข้าถึงได้จาก: <http://www.banmae.go.th/>. (วันที่สืบค้นข้อมูล: 1 ธันวาคม 2567).
- มัทนา วิบูลย์ยศศักดิ์. (ม.ป.ป.). ทำความรู้จักกับ Internet of Things. [ออนไลน์]. เข้าถึงได้จาก: <https://www.aware.co.th/th/>. (วันที่สืบค้นข้อมูล: 25 มกราคม 2568).
- ศุภรัตน์ แยมครวญ. (2566). ESP32 คืออะไร. [ออนไลน์]. เข้าถึงได้จาก: <https://byter.in.th/esp32/what-is-esp32/>. (วันที่สืบค้นข้อมูล: 1 ธันวาคม 2567).
- ศุภวิชญ์ ขวลิทธิเชษฐ. (2561). การศึกษาความเป็นไปได้ในการเลี้ยงกุ้งฝอยในบ่อพลาสติกเชิงพาณิชย์. [ออนไลน์]. เข้าถึงได้จาก: https://archive.lib.cmu.ac.th/full/T/2561/agbus91061spwtc_full.pdf. (วันที่สืบค้นข้อมูล: 1 ธันวาคม 2567).
- สุเทพ สิงห์ทอง และมาลัยวรรณ คำพล. (2561). TDS Sensor เซ็นเซอร์วัดค่าการนำไฟฟ้าของน้ำตรวจสอบคุณภาพน้ำ Analog EC Sensor. [ออนไลน์]. เข้าถึงได้จาก: <https://www.cybertice.com/>. (วันที่สืบค้นข้อมูล: 25 มกราคม 2568).
- อติราช สุขสวัสดิ์, เณริมวุฒิ น้อยอุ่นแสน, ปิยะนุช ตั้งกิตติพล, และ จักรพันธ์ ออบมา. (2566). ระบบควบคุมคุณภาพน้ำในบ่อพักกุ้งโดยใช้อินเทอร์เน็ตของสรรพสิ่ง. [ออนไลน์]. เข้าถึงได้จาก: <https://ph02.tci-thaijo.org/>. (วันที่สืบค้นข้อมูล: 1 ธันวาคม 2567).

ภาคผนวก



รูป กุ้งฝอย 20 ตัว



รูป จำนวนกุ้งที่ตายในการทดสอบประสิทธิภาพด้วยการเตรียมบ่อกุ้งฝอยที่มีปุ๋ยมูลไก่ กรวดและทราย

โค้ดต่อไปนี้เป็นโปรแกรมที่ใช้ควบคุมระบบเลี้ยงกุ้งฝอยโดยใช้บอร์ด ESP32 ซึ่งทำหน้าที่จัดการเครื่องให้อาหาร ออกซิเจน เครื่องกรองน้ำ เครื่องระบายความร้อน และเครื่องทำความร้อน รวมถึงส่งข้อมูลไป Google Sheets และ แจ้งเตือนผ่าน Telegram ด้วย

```
#include <Wire.h>
#include <DS1302.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <HTTPClient.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <ESP32Servo.h>
#include <WiFi.h>
#include <time.h>

#define ONE_WIRE_BUS 14 // DS18B20
#define TDS_PIN 35 // TDS
#define FEEDER_SERVO_PIN 13 // Servo
#define LED_ON_BOARD 2
#define OXYGEN_PUMP_PIN 25
#define FILTER_PIN 26
#define HEATER_PIN 33
#define COOLER_PIN 32
#define LIGHT_PIN 27
#define RST_RTC 23
#define DAT_RTC 22
#define CLK_RTC 21
#define OLED_SDA 16
#define OLED_SCL 17
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
#define VREF 3.3 // ค่าแรงดันอ้างอิงของ ESP32
#define ADC_RESOLUTION 4095 // ความละเอียดของ ADC (12-bit)
```

```

float TDS_MAX = 120.0;
float TEMP_MIN = 25.0;
float TEMP_MAX = 30.0;
const float HYSTERESIS = 0.3;

const char* ssid = "YourSSID";
const char* password = "YourPassword";

const char* ntpServer = "pool.ntp.org";
const long gmtOffset_sec = 25200; // GMT+7
const int daylightOffset_sec = 0;

const char* googleScriptURL = "Your URL";

// DS18B20
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
//OLED
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
// Servo for feeding
Servo feederServo;

bool foodGiven = false;
bool oxygenOffMessage = false;
bool oxygenOnMessage = false;

// RTC Module
DS1302 rtc(RST_RTC, DAT_RTC, CLK_RTC);
Time t;
unsigned long previousMillis = 0;
const unsigned long interval = 60000;

```



```

// ข้อมูล Telegram
String botToken = "Your Token"; // ใส่โทเคนที่ได้จาก BotFather
String chatID = "Your Chat ID"; // ใส่ Chat ID ของคุณ
float lastTemp = 0.00;
float lastTDS = 0.00;

void setup() {
  Serial.begin(115200);
  Wire.begin(OLED_SDA, OLED_SCL);

  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);

  display.setCursor(0, 0);
  display.println("Initializing...");
  display.display();
  // Output Pin Modes
  pinMode(OXYGEN_PUMP_PIN, OUTPUT);
  pinMode(HEATER_PIN, OUTPUT);
  pinMode(COOLER_PIN, OUTPUT);
  pinMode(FILTER_PIN, OUTPUT);

  pinMode(LIGHT_PIN, OUTPUT);
  pinMode(LED_ON_BOARD, OUTPUT);

  digitalWrite(LIGHT_PIN, HIGH);

  digitalWrite(OXYGEN_PUMP_PIN, LOW);
  digitalWrite(HEATER_PIN, HIGH);
  digitalWrite(COOLER_PIN, HIGH);
  digitalWrite(FILTER_PIN, HIGH);

```

```

// RTC Initialization
rtc.halt(false);
rtc.writeProtect(false);
connectWiFi();
setTime();
sensors.begin();
feederServo.attach(FEEDER_SERVO_PIN);
feederServo.write(0);
analogReadResolution(12);
display.clearDisplay();
sendMessage("เริ่มต้นเปิดเครื่องแล้ว");
}

void loop() {
    // Get current time from RTC
    digitalWrite(LED_ON_BOARD, LOW);
    t = rtc.getTime();
    int currentHour = t.hour;
    int currentMinute = t.min;
    char currentTime[6];
    sprintf(currentTime, "%02d:%02d", currentHour, currentMinute);
    float temperature = readTemperature();
    float tdsValue = readTDS(temperature);
    char tempMessageChar[50];
    snprintf(tempMessageChar, sizeof(tempMessageChar), "อุณหภูมิ %.2f องศา", temperature);
    String tempMessage = String(tempMessageChar);
    char tdsMessageChar[50];
    snprintf(tdsMessageChar, sizeof(tdsMessageChar), "TDS: %.2f PPM", tdsValue);
    String tdsMessage = String(tdsMessageChar);
    // Define states
    bool oxygenPumpState = true;
    bool heaterState = false;
    bool coolerState = false;
    bool filterState = false;

```

```
// Feeding Control
if ((currentHour == 6 || currentHour == 18) && !foodGiven) {
    sendMessage("ให้อาหารเรียบร้อยแล้ว");
    feed();
    foodGiven = true;
}
if (currentHour != 18) {
    foodGiven = false;
}
```

```
// Oxygen Pump Control
if (currentHour == 6 && currentMinute < 5) {
    if(!oxygenOffMessage){
        sendMessage("ปิดปั๊มก่อน พักปั๊ม 5 นาที");
        oxygenOffMessage = true;
        oxygenOnMessage = false;
    }
    digitalWrite(OXYGEN_PUMP_PIN, LOW);
    oxygenPumpState = false; // ปิดปั๊มออกซิเจน
} else {
    if(!oxygenOnMessage && oxygenOffMessage){
        sendMessage("เปิดปั๊มแล้วนะ");
        oxygenOnMessage = true;
    }
    if(oxygenOffMessage){
        oxygenOffMessage = false;
    }

    digitalWrite(OXYGEN_PUMP_PIN, HIGH);
    oxygenPumpState = true; // เปิดปั๊มออกซิเจน
}
```

```

// Heater Control
if (temperature <= TEMP_MIN ) {
  if (lastTemp + HYSTERESIS > temperature && lastTemp - HYSTERESIS > temperature){
    if(temperature != -99){
      sendMessage(tempMessage + " น้ำเย็นเกินไปแล้ว เปิดเครื่องทำความร้อนเลย");
    }
    else if (temperature == -99){
      sendMessage(tempMessage + " ไม่ได้เชื่อมต่อเซนเซอร์อุณหภูมินะ");
    }
  }
  else if (lastTemp + HYSTERESIS < temperature && lastTemp - HYSTERESIS < temperature){
    sendMessage(tempMessage + " น้ำเริ่มอุ่นขึ้นมาแล้ว");
  }
  if(temperature != -99){
    digitalWrite(HEATER_PIN, LOW);
    heaterState = true;
  }
  else if (temperature == -99){
    digitalWrite(HEATER_PIN, HIGH);
    digitalWrite(COOLER_PIN, HIGH);
    heaterState = false;
    coolerState = false;
  }
}
else if (temperature >= TEMP_MAX){
  if (lastTemp + HYSTERESIS < temperature && lastTemp - HYSTERESIS < temperature){
    sendMessage(tempMessage + " น้ำร้อนเกินไปแล้ว ระบายความร้อนออกหน่อยดีกว่า");
  }
  else if (lastTemp + HYSTERESIS > temperature && lastTemp - HYSTERESIS > temperature){
    sendMessage(tempMessage + " น้ำเริ่มเย็นขึ้นมาแล้ว");
  }
  digitalWrite(COOLER_PIN, LOW);
  coolerState = true; }

```

```

else{
    digitalWrite(HEATER_PIN, HIGH);
    digitalWrite(COOLER_PIN, HIGH);
    heaterState = false;
    coolerState = false;
}

lastTemp = temperature; // อัปเดตค่าอุณหภูมิก่อนหน้านี้

if (tdsValue >= TDS_MAX) {
    if (lastTDS + 30 < tdsValue && lastTDS - 30 < tdsValue){
        sendMessage(tdsMessage + " น้ำขุ่นเกินละ เปิดเครื่องกรองนะ");
    }
    else if (lastTDS + 30 > tdsValue && lastTDS - 30 > tdsValue){
        sendMessage(tdsMessage + " น้ำเริ่มใสขึ้นมาแล้ว");
    }
    digitalWrite(FILTER_PIN, LOW);
    filterState = true;
}
else{
    digitalWrite(FILTER_PIN, HIGH);
    filterState = false;
}

if (currentHour >= 6 && currentHour <= 18){
    digitalWrite(LIGHT_PIN, LOW);
}
else{
    digitalWrite(LIGHT_PIN, HIGH);
}

```

```

// Display data on OLED
display.clearDisplay();
display.setCursor(97, 0);
display.printf(currentTime);
display.setCursor(0, 0);
display.printf("Temp: %.2f C\n", temperature);
display.printf("TDS: %.2f ppm\n", tdsValue);

display.printf("Oxygen: %s\n", oxygenPumpState ? "ON" : "OFF");
display.printf("Heater: %s\n", heaterState ? "ON" : "OFF");
display.printf("Cooler: %s\n", coolerState ? "ON" : "OFF");
display.printf("Filter: %s\n", filterState ? "ON" : "OFF");

display.display();
//digitalWrite(LED_ON_BOARD, LOW);
unsigned long currentMillis = millis();
if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;
    digitalWrite(LED_ON_BOARD, HIGH);
    checkWifi();
    display.fillRect(0, 50, SCREEN_WIDTH, 64, BLACK);
    sendDataToGoogleSheets(currentTime, temperature, tdsValue, oxygenPumpState,
heaterState, coolerState, filterState);
}

delay(1000);
}

```

```

float readTemperature() {
    sensors.requestTemperatures();
    float temperature = sensors.getTempCByIndex(0);
    if (temperature == DEVICE_DISCONNECTED_C) {
        Serial.println("ไม่สามารถอ่านอุณหภูมิได้");
        return -99.0; // ใช้ค่าเริ่มต้นเมื่ออ่านไม่ได้
    }
    return temperature;
}

float readTDS(int temperature) {
    int analogValue = analogRead(TDS_PIN); // อ่านค่าจาก TDS Sensor
    float voltage = analogValue * VREF / ADC_RESOLUTION; // แปลงค่า ADC เป็นแรงดันไฟฟ้า
    float tdsValue = (voltage / VREF) * 1000; // แปลงแรงดันไฟฟ้าเป็นค่า TDS (mg/L หรือ ppm)
    if (temperature < 0) {
        temperature = 25;
    }
    // การชดเชยอุณหภูมิ (Temperature Compensation)
    tdsValue = tdsValue / (1.0 + 0.02 * (temperature - 25.0));
    return tdsValue;
}

```

```

void setTime(){
  // Sync RTC with NTP
  configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);
  struct tm timeinfo;
  if (getLocalTime(&timeinfo)) {
    rtc.setTime(
      timeinfo.tm_hour,
      timeinfo.tm_min,
      timeinfo.tm_sec
    );
    Serial.println("RTC time updated from NTP!");
    display.setCursor(0, 50);
    display.println("Time Update!");
    display.display();
  } else {
    Serial.println("Failed to get time from NTP");
  }
}

```

```

void checkWifi(){
  if (WiFi.status() != WL_CONNECTED) {
    WiFi.begin(ssid, password);
    display.setCursor(0, 50);
    display.println("Connecting to Wi-Fi..");
    display.display();

  }
  else if (WiFi.status() == WL_CONNECTED){
    setTime();
  }
}

```



```

void connectWiFi() {
  display.clearDisplay();
  display.setCursor(0, 0);
  display.println("Connecting to Wi-Fi...");
  display.display();
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  int attempts = 0;

  while (WiFi.status() != WL_CONNECTED && attempts < 20) {
    WiFi.begin(ssid, password);
    digitalWrite(LED_ON_BOARD, HIGH);
    delay(250);
    digitalWrite(LED_ON_BOARD, LOW);
    delay(250);
    Serial.print(".");

    // แสดงความคืบหน้า
    display.fillRect(0, 10, SCREEN_WIDTH, 10, BLACK);
    display.setCursor(0, 10);
    display.printf("Attempt %d/20", attempts + 1);
    display.display();

    attempts++;

  }

  if (WiFi.status() == WL_CONNECTED) {
    Serial.println("\nWi-Fi connected!");
    display.clearDisplay();
    display.setCursor(0, 0);
    display.println("Wi-Fi connected!");
    display.display();
  }
}

```

```

} else {
    Serial.println("\nFailed to connect.");
    display.clearDisplay();
    display.setCursor(0, 0);
    display.println("Failed to connect.");
    //display.println("Restarting...");
    display.display();
    delay(1000); // แสดงข้อความสักครู่ก่อนรีบูต
    //ESP.restart(); // รีบูตบอร์ด
}
}

// ให้อาหาร
void feed() {
    feederServo.write(90);
    delay(500);
    feederServo.write(0);
}

```

```
// ส่งข้อมูลไปยัง Google Sheets
void sendDataToGoogleSheets(String time, float temperature, float tds, bool
oxygenPumpState, bool heaterState, bool coolerState, bool filterState) {
  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;
    // Prepare JSON payload
    String jsonPayload = "{\"time\":\"" + time +
      "\",\"temperature\":\"" + String(temperature) +
      "\",\"tds\":\"" + String(tds) +
      "\",\"oxygenPumpState\":\"" + (oxygenPumpState ? "true" : "false") +
      "\",\"heaterState\":\"" + (heaterState ? "true" : "false") +
      "\",\"coolerState\":\"" + (coolerState ? "true" : "false") +
      "\",\"filterState\":\"" + (filterState ? "true" : "false") + "\"}";
    http.begin(googleScriptURL);
    http.addHeader("Content-Type", "application/json");
    int httpResponseCode = http.POST(jsonPayload);
    if (httpResponseCode > 0) {
      Serial.println("Data sent successfully to Google Sheets.");
      Serial.println(http.getString());
    } else {
      Serial.print("Error sending data: ");
      Serial.println(httpResponseCode);
    }
    display.setCursor(0, 50);
    display.println("Sent to Sheet");
    display.display();
    http.end();
  } else {
    Serial.println("Wi-Fi not connected. Unable to send data.");
  }
}
```

```

void sendMessage(String message) {
    if(WiFi.status() == WL_CONNECTED) {
        HTTPClient http;

        String url = "https://api.telegram.org/bot" + botToken + "/sendMessage?chat_id=" + chatID
+ "&text=" + message;
        http.begin(url);
        int httpResponseCode = http.GET();

        if(httpResponseCode > 0) {
            Serial.println("ส่งข้อความสำเร็จ: " + String(httpResponseCode));
        } else {
            Serial.println("ไม่สามารถส่งข้อความได้, รหัสข้อผิดพลาด: " + String(httpResponseCode));
        }
        http.end();
    } else {
        Serial.println("ไม่ได้เชื่อมต่อ Wi-Fi");
    }
}

```

โค้ดต่อไปนี้เป็นโค้ด Google API ที่นำค่าจากบอร์ด ESP32 ที่จะส่งไปเก็บที่ Google Sheets

```

function doPost(e) {
    try {
        // Spreadsheet ID ที่ได้รับ
        const spreadsheetId = '15k59_eiUn-dCHsFXYSLBXVW75ewh5JZTnbffYCN1JiU';

        // รับข้อมูลจาก ESP32
        const data = JSON.parse(e.postData.contents);
        const date = new Date();
        const today = Utilities.formatDate(date, "GMT+7", "yyyy-MM-dd"); // ชื่อซีตเป็นวันที่ปัจจุบัน

        // เปิด Google Sheets ตาม ID ที่กำหนด
        const spreadsheet = SpreadsheetApp.openById(spreadsheetId);
        let sheet = spreadsheet.getSheetByName(today);
    }
}

```

```

// ถ้าไม่มีชีตวันนี้ ก็สร้างใหม่
if (!sheet) {
    sheet = spreadsheet.insertSheet(today);
    sheet.appendRow(["Time", "Temperature", "TDS", "Oxygen Pump State", "Heater State",
"Cooler State", "Filter State"]); // หัวตาราง
}

// ดึงข้อมูลที่ส่งมาจาก ESP32
const time = data.time || "00:00";
const temperature = data.temperature || 0;
const tds = data.tds || 0;
const oxygenPumpState = data.oxygenPumpState ? "ON" : "OFF";
const heaterState = data.heaterState ? "ON" : "OFF";
const coolerState = data.coolerState ? "ON" : "OFF";
const filterState = data.filterState ? "ON" : "OFF";

// เพิ่มข้อมูลลงในชีตวันนี้
sheet.appendRow([time, temperature, tds, oxygenPumpState, heaterState, coolerState,
filterState]);

return ContentService.createTextOutput('Success');
} catch (error) {
    return ContentService.createTextOutput('Error: ' + error.message);
}
}

```