

การพัฒนา API เพื่อลดความซับซ้อนในการพัฒนาแอปพลิเคชัน
สำหรับอุปกรณ์เซนเซอร์โนดที่ทำงานบนแพลตฟอร์มอาร์ดูโน

Development of an API to reduce complexity in application developments
on Arduino-based sensor nodes

ขนิษฐา แซ่ลิ้ม^{1*} วีระศักดิ์ ชื่นตา¹ และ นิธิฐิตา เชิดชู¹
คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครปฐม¹

Khanittha Saelim^{1*} Weerasak Cheunta¹ and Nitthita Chirdchoo¹
Faculty of Science and Technology, Nakhon Pathom Rajabhat University¹
E-mail : kanit@webmail.npru.ac.th^{1*} weerasak@webmail.npru.ac.th¹ ,
nitthita@webmail.npru.ac.th¹

บทคัดย่อ

การวิจัยครั้งนี้มีวัตถุประสงค์เพื่อออกแบบและเพิ่มความเป็นมิตรกับผู้ใช้งานในส่วนของการพัฒนาแอปพลิเคชันบนเซนเซอร์โนดที่เรียกว่า Simple-API ทั้งนี้เพื่อให้เกิดความง่ายในการเรียกใช้งาน โดยการพัฒนางานจะอิงกับโครงสร้างการทำงานของเซนเซอร์โนดซึ่งประกอบด้วย ส่วนประมวลผล ส่วนตรวจวัดและสั่งการ ส่วนสื่อสารไร้สาย และส่วนควบคุมพลังงานและแบตเตอรี่ และใช้หลักการเขียนโปรแกรมเชิงวัตถุร่วมกับรูปแบบการสร้างคลังโปรแกรมในแพลตฟอร์มอาร์ดูโนด้วยภาษาโปรแกรมภาษาซีพลัสพลัส ผู้วิจัยได้สร้างฮาร์ดแวร์เซนเซอร์โนดโดยใช้หน่วยประมวลผล Arduino Mega2560 และทำการติดตั้งและทดสอบหาประสิทธิภาพ Simple-API ที่พัฒนาขึ้น พบว่า สามารถใช้งานได้ง่าย การสั่งการด้วยคำสั่งสั้นๆ ที่จัดเตรียมไว้ ทั้งนี้เนื่องจากการสร้างและจัดกลุ่มฟังก์ชันการทำงานโดยซ่อนการเขียนโปรแกรมระดับล่างและการเข้าถึงเรจิสเตอร์ของอุปกรณ์ที่ผู้ใช้ไม่จำเป็นต้องปรับเปลี่ยนในการใช้งาน มีเสถียรภาพการทำงานต่อเนื่อง ได้ทำการทดสอบในสภาวะสิ่งแวดล้อมที่ไม่เหมาะสมทั้งอุณหภูมิ ความชื้น รวมถึงระบบไฟฟ้าในสถานที่จริง พบว่า ตลอดระยะเวลากว่า 3 เดือนเซนเซอร์โนดที่ติดตั้ง Simple-API สามารถทำงานได้อย่างต่อเนื่อง ทั้งนี้ด้วยความช่วยเหลือจากการทำงานในส่วนของการ watchdog timer ที่ออกแบบให้ทำงานคู่ขนานกับทุก Simple-API รวมถึงการป้องกันการความผิดพลาด ในการเรียกใช้งานและระหว่างอุปกรณ์ทำงาน

Abstract

The objective of this work is to design a user-friendly API to be used with an Arduino-based sensor mote applications, called Simple-API. The API is designed based on the sensor mote architecture which includes processing, sensing and controlling, communicating and battery and power managing units. Moreover, the software is developed with OOP C++ together with Arduino standard libraries. Simple-API is easy to use, able to reduce the number of lines of code needed

by reducing the complexity in handling the low-level interfacing. To evaluate the performance, we implement an application using Simple-API on a sensor mote with Arduino Mega2560. The sensor mote is then deployed and tested for 3 months. During the experiment, we have also implemented the watchdog timer to work in parallel with the Simple-API, in order to avoid any freeze of both the hardware and software caused by electrical fluctuation. As a result, it is found that the sensor mote implemented with our proposed Simple-API can continuously run throughout the crop without any malfunction.

Keywords : Sensor Node, Sensor Network, API design

1. บทนำ

ปัจจุบันความต้องการใช้งานเครือข่ายเซนเซอร์ไร้สายมีเพิ่มขึ้นและหลากหลายรูปแบบ เช่น การนำไปใช้ในระบบฟาร์มเพาะเลี้ยงสัตว์น้ำอัจฉริยะ ระบบป้องกันและเตือนภัยการเกิดไฟฟ้า ระบบติดตามสัตว์เลี้ยง ระบบการจราจรอัจฉริยะ เป็นต้น [1] องค์ประกอบหลักของการพัฒนาระบบเครือข่ายเซนเซอร์ไร้สายข้างต้น ประกอบไปด้วย ส่วนของฮาร์ดแวร์ซึ่งก็คือ ตัวเซนเซอร์โนด และส่วนของซอฟต์แวร์ซึ่งก็คือ การพัฒนาชุดโปรแกรมแอปพลิเคชันเพื่อให้เซนเซอร์โนดทำงานได้ตามที่ได้ออกแบบไว้ ในการพัฒนาชุดโปรแกรมแอปพลิเคชันนั้นส่วนใหญ่จะใช้ภาษา C หรือ C++ [2] การพัฒนาชุดโปรแกรมที่ต้องทำหน้าที่ให้สอดคล้องกับโครงสร้างและการทำงานของเซนเซอร์โนด ที่สามารถแบ่งออกเป็น 4 โมดูล [3] นั่นคือ (1) โมดูลสำหรับการสื่อสาร (2) โมดูลสำหรับการตรวจวัดค่าจากสิ่งแวดล้อม (3) โมดูลสำหรับการประมวลผล และ (4) โมดูลสำหรับระบบพลังงานของเซนเซอร์โนด

การพัฒนาชุดโปรแกรมหรือแอปพลิเคชันเพื่อสั่งการให้ตัวเซนเซอร์โนดทำงาน ผู้พัฒนาโปรแกรมจะต้องมีความเข้าใจการทำงานทั้งในส่วนของฮาร์ดแวร์ที่ประกอบด้วย ตัวเซนเซอร์ อุปกรณ์สื่อสาร แหล่งจ่ายพลังงาน การอินเทอร์เฟสอื่นๆ เช่น ระบบฐานเวลา และการจัดการหน่วยความจำ เป็นต้น และส่วนของอัลกอริทึมการทำงานของเซนเซอร์โนด จึงจะสามารถพัฒนาชุดโปรแกรมออกมาได้ตามต้องการ ปัญหาที่มักเกิดขึ้นในขั้นตอนของการพัฒนาแอปพลิเคชันก็คือ ผู้พัฒนาแอปพลิเคชันมักมีความเชี่ยวชาญในการเขียนโปรแกรม แต่มักขาดทักษะและการทำความเข้าใจในการทำงานของอุปกรณ์ฮาร์ดแวร์ ทำให้การเขียนโปรแกรมอินเทอร์เฟสกับส่วนของฮาร์ดแวร์ซึ่งบ่อยครั้งต้องมีการจัดการในส่วนของหน่วยความจำ และ hardware register นั้นมีความยุ่งยากและซับซ้อน ในมุมมองของผู้พัฒนาโปรแกรมส่งผลให้การพัฒนาแอปพลิเคชันมีความล่าช้า และอาจไม่ประสบผลสำเร็จตามที่ได้ออกแบบไว้

จากงานพัฒนาแพลตฟอร์ม (platform) เปิดสำหรับเครือข่ายเซนเซอร์ไร้สาย [6] ของมหาวิทยาลัยอัลบอร์ ประเทศเดนมาร์ก ออกแบบเซนเซอร์โนดเพื่อใช้ในห้องปฏิบัติการ โดยใช้ไมโครคอนโทรลเลอร์ dsPIC30f3013 เป็นหน่วยประมวลผล พบว่า การดาวน์โหลดโปรแกรมต้องใช้เครื่องโปรแกรม PICkit 2 ซึ่งยุ่งยาก ละที่สำคัญมีการจำกัดความสามารถในการทำงานสำหรับโปรแกรมรุ่นที่อนุญาตให้ใช้งานฟรี หากต้องการเขียนโปรแกรมที่มีขนาดใหญ่ ซับซ้อนขึ้นผู้พัฒนาต้องเสียค่าใช้จ่ายเพิ่ม

เซนเซอร์โนด PSU Mote [7] ที่ถูกพัฒนาขึ้นโดยคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ เลือกใช้ไมโครคอนโทรลเลอร์ LPC2103 ของบริษัท NXP เป็นหน่วยประมวลผลหลัก การดาวน์โหลดโปรแกรมสามารถทำได้ผ่านช่องอนุกรมโดยใช้โปรแกรม Flash Magic

เซนเซอร์โนด Waspote [8] ถูกพัฒนาขึ้นในปี ค.ศ. 2009 โดยบริษัท Libelium มีการดำเนินงานในเชิงพาณิชย์แล้ว จุดเด่นคือการเปิดเผยโค้ดต้นฉบับ ต่อยอดแนวคิดจากการใช้งานอาดูอิน (Arduino) จากการทดสอบใช้งาน Waspote ผู้วิจัยพบว่า การโปรแกรมเช่นสั่งการขับอุปกรณ์เอาต์พุต ซึ่งเป็นโปรแกรมพื้นฐานนั้นมีการเรียกใช้ฟังก์ชันการทำงานจำนวนมาก ส่งผลให้เกิดความสับสนได้ในการใช้งาน และที่สำคัญระยะเวลาในการแปลโปรแกรม (compile) การดาวน์โหลดโปรแกรมลงบน Waspote ใช้เวลาค่อนข้างมาก

จากความสำคัญปัญหาและแนวทางดังกล่าวข้างต้น ผู้วิจัยจึงมีแนวคิดในการพัฒนา API ที่ง่ายต่อการพัฒนาโปรแกรมบนเซนเซอร์โนด ช่วยให้การพัฒนาแอปพลิเคชันในระบบเครือข่ายเซนเซอร์ไร้สายสามารถเป็นไปได้อย่างง่ายดาย และมีความรวดเร็ว API ที่ออกแบบต้องสามารถทำงานได้บน Arduino IDE ซึ่งเป็นรูปแบบการพัฒนาโปรแกรมที่ได้รับความนิยมมากในปัจจุบัน และในการออกแบบ API จะได้ยกรณีกฎการศึกษาเกี่ยวกับการเซนเซอร์ในการตรวจวัดคุณภาพน้ำเพื่อการเพาะเลี้ยงสัตว์น้ำเศรษฐกิจของประเทศ ผลจากการวิจัยเชื่อว่าจะเป็นกระตุ้นให้เกิดการพัฒนาแอปพลิเคชันบนเครือข่ายเซนเซอร์ไร้สายในวงกว้างต่อไป

2. วิธีการวิจัย (Methodology)

การวิจัยครั้งนี้เริ่มจากการกำหนดเซนเซอร์ที่มีความสำคัญในการเพาะเลี้ยงสัตว์น้ำเศรษฐกิจของประเทศ จากนั้นดำเนินการออกแบบ API ออกแบบฮาร์ดแวร์เพื่อติดตั้ง API และทำการทดสอบการทำงานของ API ทั้งในห้องปฏิบัติการและสถานที่จริง โดยมีขั้นตอนและรายละเอียดดังนี้

2.1 การกำหนดเซนเซอร์ที่เกี่ยวข้องในการออกแบบ API

ในงานด้านการเกษตรเกี่ยวกับการเพาะเลี้ยงสัตว์น้ำซึ่งถือว่าเป็นเศรษฐกิจของประเทศ เช่น ปลา กุ้ง ปังจี้สำคัญในการเพาะเลี้ยงให้ประสบผลสำเร็จได้แก่คุณภาพน้ำ โดยเฉพาะปริมาณออกซิเจนละลาย (DO) ค่าความเป็นกรดต่าง (pH) ค่าความนำ (EC) และอุณหภูมิของน้ำที่ต้องเหมาะสมกับสัตว์แต่ละประเภท การออกแบบ Simple-API และเซนเซอร์โนด ให้ตอบสนองกับความต้องการดังกล่าว จึงเป็นสิ่งจำเป็นและมีประโยชน์อย่างยิ่ง

เซนเซอร์โนดที่ในงานวิจัยเรื่องการตรวจวัดคุณภาพแหล่งน้ำ [4] ในประเทศอินเดีย นำเสนอการประยุกต์ใช้เครือข่ายเซนเซอร์ไร้สายสำหรับตรวจวัดคุณภาพในแหล่งน้ำธรรมชาติ ส่วนประมวลผลใช้ไมโครคอนโทรลเลอร์ตระกูล PIC16 รุ่น 8 บิต จากบริษัทไมโครชิพ (Microchip) เชื่อมต่อกับเซนเซอร์ประเภทแอนะล็อกสำหรับตรวจวัดค่าพีเอช การสื่อสารเลือกใช้เครื่องรับส่งสัญญาณวิทยุที่ใช้โปรโตคอลชิป

แนวทางข้างต้นผู้วิจัยนำมากำหนดเซนเซอร์เพื่อพัฒนา Simple-API เพื่อนำไปประยุกต์ใช้ในการตรวจวัดคุณภาพน้ำ ได้แก่ เซนเซอร์วัด DO pH และ EC ในน้ำ ของบริษัท Atlas Scientific Biology Technology ดังรูปที่ 1 - รูปที่ 3 ซึ่งมาพร้อมแผงวงจรชั้นกลางระหว่างเซนเซอร์กับหน่วยประมวลผล ใช้การสื่อสารแบบอนุกรม RS232 หรือแบบ I2C การใช้งานสามารถทำได้โดยใช้คำสั่งในรูปแบบรหัสแอสกี (ASCII code)



รูปที่ 1 ชุดเซนเซอร์วัดค่าออกซิเจนละลายในน้ำ
(DO Sensor)



รูปที่ 2 ชุดเซนเซอร์วัดค่าความเป็นกรด-ด่างในน้ำ
(pH Sensor)



รูปที่ 3 ชุดเซนเซอร์วัดค่าการนำไฟฟ้าในน้ำ
(EC Sensor : Electrical Conductivity)



รูปที่ 4 เซนเซอร์ตรวจวัดอุณหภูมิ DS18B20

ส่วนของเซนเซอร์ตรวจวัดอุณหภูมิ (temperature sensor) ดังรูปที่ 4 ซึ่งผู้วิจัยเลือกใช้ประเภทดิจิทัล DS18B20 จากบริษัท Dallas Semiconductor ตรวจวัดอุณหภูมิได้ในช่วง 55 – 125 องศาเซลเซียส สามารถกำหนดค่าความละเอียดในการวัดในระดับทศนิยมสามตำแหน่ง ใช้เวลาในการตรวจวัดสูงสุดไม่เกิน 750 มิลลิวินาที ต่อครั้ง การเชื่อมต่อกับหน่วยประมวลผลผ่านบัส 1-Wire ใช้แรงดันไฟเลี้ยง 3 – 5.5 โวลต์

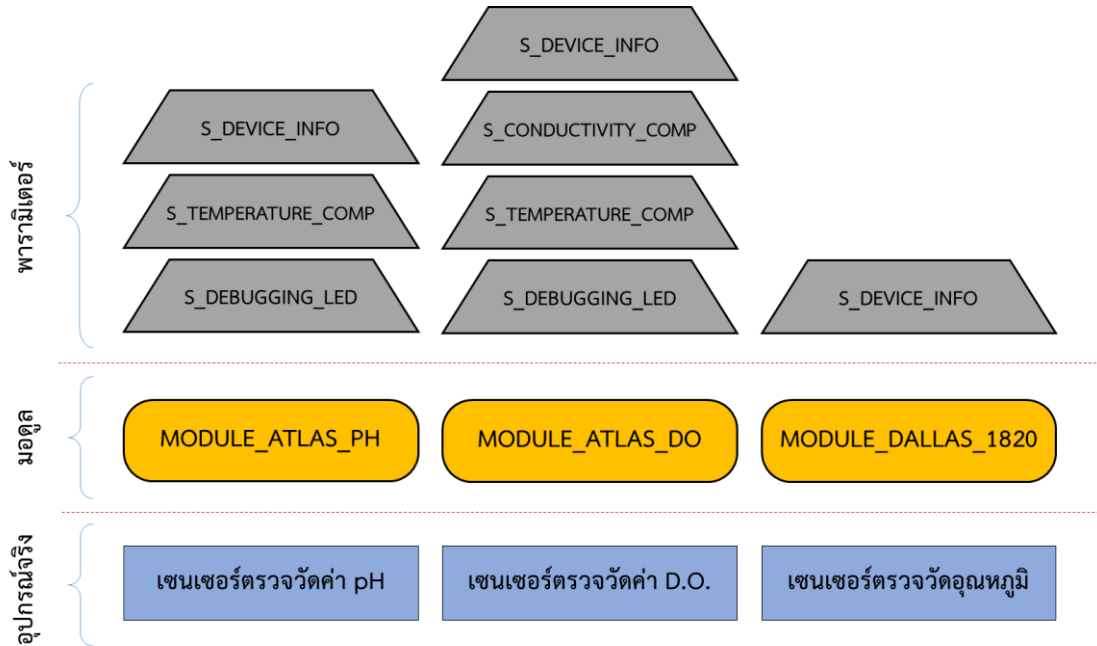
2.2 การออกแบบและพัฒนาส่วนของ Simple-API

เซนเซอร์โนด WESENSE [5] มีการออกแบบและสร้างขึ้นจากแนวคิดการแบ่งระบบการทำงานในเซนเซอร์โนดออกเป็นระบบย่อยๆ ได้แก่ ระบบประมวลผล ระบบสื่อสารไร้สาย ระบบพลังงาน และระบบตรวจวัด โดยพัฒนาขึ้นมาบนแผงวงจรที่แยกจากกัน การใช้งานจะนำแผงวงจรของแต่ละระบบซ้อนกันเป็นชั้นๆ WESENSE ใช้ไมโครคอนโทรลเลอร์ MSP430G2955 เป็นหน่วยประมวลผลหลัก

ส่วนของโปรแกรมประยุกต์ Simple-API ผู้วิจัยมีแนวคิดในการออกแบบและพัฒนาขึ้นโดยอ้างอิงกับโครงสร้างการทำงานของเซนเซอร์โนด ส่วนประมวลผล ส่วนตรวจวัดและสั่งการ ส่วนสื่อสารไร้สาย และส่วนควบคุมพลังงานและแบตเตอรี่ โดยใช้หลักการเขียนโปรแกรมเชิงวัตถุ (object oriented programming - OOP) ร่วมกับรูปแบบการสร้างคลังโปรแกรมในแพลตฟอร์มมาดูลิโนด้วยภาษาโปรแกรมภาษาซี ++ การทำงานทั้งหมดจะถูกนำมาสร้างเป็นคลาส ได้แก่ คลาส ProcessingSystemC, คลาส SensorAndActuatorC, คลาส WirelessCommunicationC, และ PowerControlSystemC ภายในแต่ละคลาสนำเสนอแนวคิดการสร้างสัญลักษณ์เรียกว่า มอดูล แทนอุปกรณ์ที่นำมาใช้งานในส่วนการทำงาน ร่วมกับสัญลักษณ์เรียกว่า พารามิเตอร์ แทน

ค่าที่ใช้สำหรับปรับแต่งการทำงานหรือค่าที่อ่านได้ภายในอุปกรณ์ คลาสของส่วนต่อประสานโปรแกรมประยุกต์ Simple-API

ตัวอย่างคลาส SensorAndActuatorC ถูกกำหนดและออกแบบให้ทำหน้าที่ตรวจวัดคุณภาพน้ำและสั่งการออกแบบมอดูลและพารามิเตอร์ที่ใช้งานภายในคลาสโครงสร้าง ดังรูปที่ 5



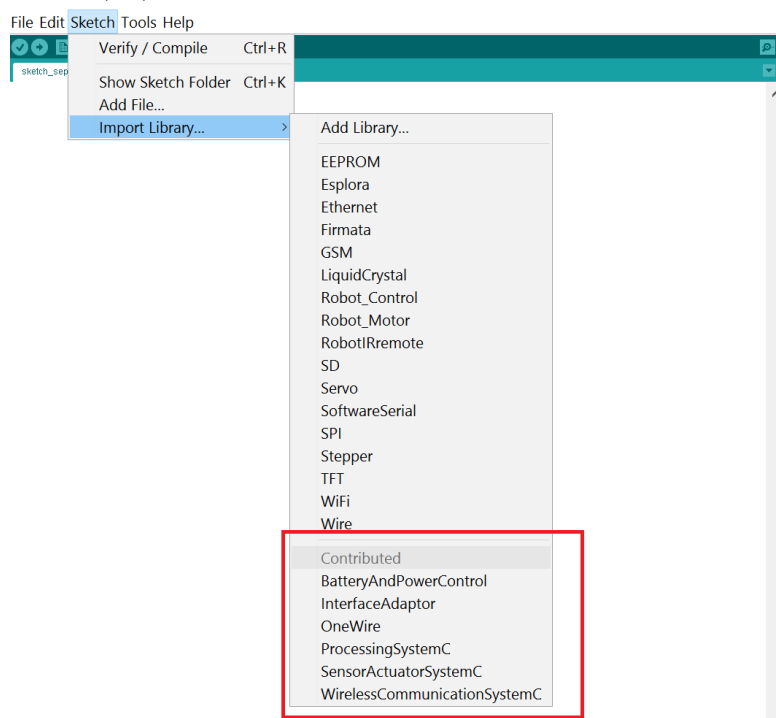
รูปที่ 5 โครงสร้างการออกแบบภายในคลาส SensorAndActuatorC

ภายในมอดูล MODULE_ATLAS_PH ทำหน้าที่เปรียบเสมือนชุดอุปกรณ์เซนเซอร์ตรวจวัดค่า pH ภายในออกแบบให้มีพารามิเตอร์สำหรับใช้งานทั้งหมด 3 ตัว ในตารางที่ 1

ตารางที่ 1 รายชื่อพารามิเตอร์ภายในมอดูล MODULE_ATLAS_PH

พารามิเตอร์	รายละเอียด
S_DEBUGGING_LED	เปิดหรือปิดการใช้งานหลอดแอลอีดีเพื่อตรวจสอบการทำงานบนวงจรเซนเซอร์
S_TEMPERATURE_COMP	ค่าอุณหภูมิของสารละลายที่ทำการวัดเพื่อใช้เป็นข้อมูลในการคำนวณหาค่าพีเอช ไม่จำเป็นต้องกำหนดค่าอุณหภูมิทุกครั้งที่ทำกรวัดหากไม่ได้สั่งปิดการจ่ายพลังงานไฟฟ้าให้กับเซนเซอร์
S_DEVICE_INFO	ข้อมูลหมายเลขรุ่นและวันที่ของเฟิร์มแวร์ภายในวงจรเซนเซอร์

จากการออกแบบ Simple-API ข้างต้นแล้วเสร็จ ผู้วิจัยได้นำมาติดตั้งเข้ากับ Arduino IDE ซึ่งสามารถเรียกใช้งานได้ผ่านเมนู Sketch -> Import Library ดังรูปที่ 6



รูปที่ 6 รายละเอียดการติดตั้งไลบรารีและตัวอย่างโปรแกรมใช้งาน

```

/*----- STANDARD ARDUINO LIBRARIES -----*/
#include <EEPROM.h>
#include <OneWire.h>
#include <Wire.h>
#include <SPI.h>
#include <SoftwareSerial.h>
/*----- Simple-API LIBRARIES -----*/
#include <InterfaceAdaptor.h>
#include <ProcessingSystemC.h>
#include <PowerControlSystemC.h>
#include <WirelessCommunicationSystemC.h>
#include <SensorActuatorSystemC.h>
void setup(){
    SensorObject.begin(MODULE_ATLAS_DO, HW_SERIAL_1);
}
void loop() {
    int8_t result = SensorObject.readSensorData(MODULE_ATLAS_DO, &DOValue,
        sizeof(DOValue));
}
    
```

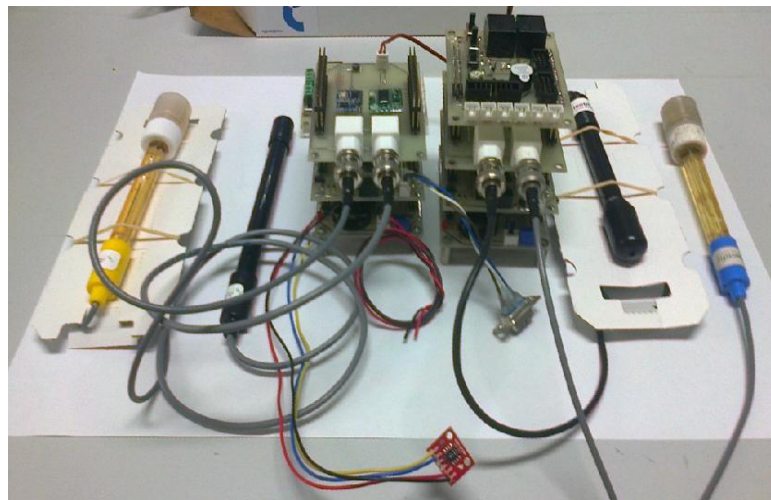
รูปที่ 7 รายละเอียดโปรแกรมการติดตั้งไลบรารีพื้นฐานและ Simple-API ที่พัฒนาขึ้น

จากรูปที่ 7 เป็นการเลือกติดตั้ง Simple-API ที่พัฒนาขึ้นทั้ง 4 API และการติดตั้งไลบรารีพื้นฐาน STANDARD ARDUINO LIBRARIES อื่นที่จำเป็น จากการทดสอบพบว่า พื้นที่หน่วยความ Global variables จำนวน 488 byte

ตัวอย่างการใช้งานเริ่มจาก คำสั่ง SensorObject.begin ใน void setup เป็นการบอกเซนเซอร์โนดให้ การเปิดการสื่อสารพอร์ตอนุกรมหมายเลข 1 สำหรับเซนเซอร์วัดออกซิเจน และในส่วนของ void loop คำสั่ง SensorObject.readSensorData เป็นการอ่านค่า DO จากเซนเซอร์

2.3 การออกแบบส่วนของฮาร์ดแวร์เพื่อการทดสอบประสิทธิภาพของ Simple-API ที่พัฒนาขึ้น

ในรูปที่ 8 เป็นชุดเซนเซอร์โนดที่พัฒนาขึ้นประกอบพร้อมนำไปใช้งานเพื่อทดสอบหาประสิทธิภาพของ API ทั้งในห้องปฏิบัติการและสถานที่จริง ประกอบกันเป็นชั้นรวมทั้ง 4 ชั้น ขนาดกว้างยาว 6x8 เซนติเมตร แต่ละชั้น ประกอบด้วยชั้นล่างสุดชั้นที่ 1 เป็นชั้นของวงจรชุดจ่ายไฟ (วัดพลังงาน ควบคุมและจ่ายแรงดัน) ชั้นที่ 2 เป็นชุดของ บอร์ดประมวลผล (ATMega2560) และระบบสื่อสาร (แผงวงจรรับส่งสัญญาณวิทยุ บูลทูธ แผงวงจรชิป) ชั้นที่ 3 เป็นชั้นของเซนเซอร์ (DO pH อุณหภูมิ) และชั้นบนสุดชั้นที่ 4 เป็นชั้นของอินพุทเอาต์พุตและอินเตอร์เฟซอื่นๆ แต่ละชั้นสามารถถอดแยกจากกันได้ เหตุผลเพื่อการใช้งานที่แตกต่างกัน บางกรณีอาจจะไม่จำเป็นต้องมีชั้นที่ 4 ก็สามารถถอดออกได้ เป็นต้น



รูปที่ 8 แผงวงจรเซนเซอร์โนดที่พัฒนาขึ้นเพื่อใช้ทดสอบ Simple-API

3. ผลการวิจัย (Results)

3.1 ประสิทธิภาพด้านความเป็นมิตรกับผู้ใช้

หัวใจหลักสำคัญของการวิจัยในครั้งนี้คือการออกแบบและเพิ่มความเป็นมิตรกับผู้ใช้ในส่วน ของซอฟต์แวร์ที่ใช้ในการพัฒนาแอปพลิเคชันบนเซนเซอร์โนด นั้นหมายถึงความง่ายสะดวกในการเรียกใช้งาน จากตารางที่ 2 ยกตัวอย่างการเขียนโปรแกรมเพื่ออ่านค่า DO โดยเรียกเพียงคำสั่งเดียว API ก็จะทำให้จนให้ค่า DO กลับมายังผู้ใช้

กรณีเขียนโปรแกรมสื่อสารกับเซนเซอร์เพื่ออ่านค่า DO โดยตรง ผู้เขียนจะต้องทำการส่งคำสั่งเพื่ออ่านค่า ออกซิเจนไปยังมอดูลเซนเซอร์ void single_reading_mode() { } รายละเอียดดังรูปที่ 9 และต้องทำการเขียน โปรแกรมเพื่อรับค่ามอดูลเซนเซอร์ void device_response_data() { } รายละเอียดดังรูปที่ 10

จากกรณีตัวอย่างจะเห็นว่า ผู้พัฒนาโปรแกรมโดยเรียกใช้ Simple-API เพียงทำการติดตั้งไลบรารีพื้นฐาน และเรียกใช้คำสั่งดังตารางที่ 2 ก็จะได้รับค่า DO เพื่อนำไปประมวลผลต่อไป ส่วนการโปรแกรมโดยเรียกฟังก์ชันจาก โปรแกรมขับอุปกรณ์โดยตรงนั้น ผู้พัฒนาโปรแกรมจำเป็นต้องมีความรู้และประสบการณ์ในการเขียนโปรแกรมเป็น อย่างดี ตั้งแต่เรื่องการประกาศและจัดการเกี่ยวกับตัวแปรแบบอาเรียรี รหัสแอสกี้โค้ด การเขียนโปรแกรมวนรอบ การ รับส่งข้อมูลผ่านพอร์ตอนุกรม RS232 ตลอดจนเทคนิคการโปรแกรม เช่น การป้องกันการติดในลูป การป้องกัน ข้อมูลขยะในอาเรียรี เป็นต้น

ตารางที่ 2 เปรียบเทียบการเรียกใช้งานอุปกรณ์โดยใช้ Simple-API และการเรียกใช้โดยตรงกรณี DO Sensor

การเรียกฟังก์ชันของ SIMPLE-API	การเรียกฟังก์ชันจากโปรแกรมขับอุปกรณ์โดยตรง
SENSOROBJ.READSENSORDATA(MODULE_ATLAS_DO, &DATA, SIZEOF(DATA));	void single_reading_mode() { } void device_response_data() { } // CPU 86 Code C

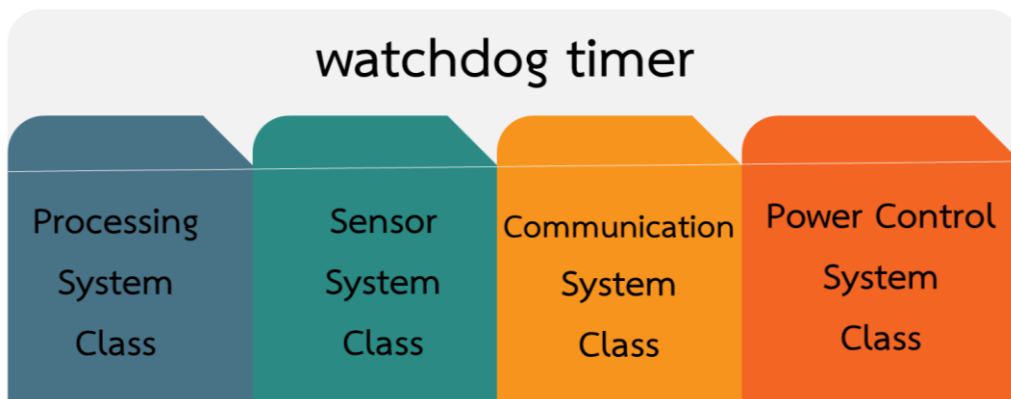
```
void single_reading_mode()
{
    int len;
    memset(cmd,0,49);
    strcpy(cmd,"R");
    len = strlen(cmd);
    for(i=0; i<len; i++)
    {
        SerPort_SendByte(COM1,cmd[i]);
    }
    SerPort_SendByte(COM1,13);
}
```

```
void device_response_data()
{
    unsigned int count=0;
    memset(response,0,49);
    c=0; i=0;
    do {
        if(SerPort_Avail(COM1))
        {
            SerPort_RecvByte(COM1,&c);
            if (c==13) { //cout<<"\n === data_com1
:=<<response<<"==";
                //i=0;
                //memset(data,0,9);
            }
            else { response[i] = c;
                i++; }
            count=0; }
        count++;
    } while ( (!kbhit()) && (count<50000)); // && (c!=13) );
    cout<<"\n === Do data_com1 :=<<response<<"=="; }
```

รูปที่ 9 รายละเอียดโปรแกรมการส่งคำสั่งเพื่ออ่านค่าออกซิเจนโดยตรงจากมอดูล DO Sensor

รูปที่ 10 รายละเอียดโปรแกรมการรับค่าออกซิเจนโดยตรงจากมอดูล DO Sensor

นอกจากสะดวกและง่ายในการใช้งานแล้ว เสถียรภาพก็เป็นสิ่งสำคัญ Simple-API ที่ออกแบบมีการผู้วิจัย ได้พิจารณาถึงเหตุการณ์ที่ไม่ปกติ โดยออกแบบส่วนของการรีเซตหน่วยประมวลผลกรณีเกิดการทำงานค้าง ที่เรียกว่า watchdog timer เข้าไว้ในทุกส่วนของ API ดังรูปที่ 11



รูปที่ 11 การติดตั้ง watchdog timer ครอบคลุมส่วนการทำงานของ Simple-API

ในการนี้เพื่อป้องกันการได้ข้อมูลที่ไม่ถูกต้อง ในแต่ละคำสั่งที่สำคัญจะมีการคืนค่าเพื่อแสดงสถานะการทำงาน ในตารางที่ 3

ตารางที่ 3 ความหมายและการตรวจสอบแก้ไขของรหัสแสดงผลการทำงาน

รหัสตัวเลข	ความหมาย	การตรวจสอบแก้ไข
0	ทำงานสำเร็จไม่มีความผิดพลาดเกิดขึ้น	-
1	ทำงานไม่สำเร็จไม่พบมอดูล	ตรวจสอบชื่อมอดูลที่เรียกใช้
2	ทำงานไม่สำเร็จไม่สามารถใช้คำสั่งกับมอดูลกับนี้ได้	ตรวจสอบมอดูลว่าอนุญาตให้ใช้คำสั่งงานใดบ้าง
3	ทำงานไม่สำเร็จไม่พบพารามิเตอร์ที่เรียกสำหรับมอดูลนี้	ตรวจสอบชื่อพารามิเตอร์ของมอดูล
4	ทำงานไม่สำเร็จค่าหรือขนาดของพารามิเตอร์ไม่ถูกต้อง	ตรวจสอบค่าที่ใช้งานหรือขนาดความยาวของพารามิเตอร์
-1	ทำงานไม่สำเร็จอุปกรณ์เชื่อมต่อไม่ตอบสนองในเวลาที่กำหนด	ตรวจสอบอุปกรณ์ที่นำมาเชื่อมต่อ
-2	ทำงานไม่สำเร็จการเชื่อมต่อผิดพลาด	ตรวจสอบการเชื่อมต่อหรือ

3.2 ประสิทธิภาพในการทำงานต่อเนื่อง

เพื่อทดสอบหาเสถียรภาพการทำงานต่อเนื่อง ผู้วิจัยได้ทำการติดตั้ง Simple-API บนอุปกรณ์เซนเซอร์โนด แล้วนำไปทดสอบติดตั้งในบ่อเลี้ยงกุ้งของเกษตรกร ดังรูปที่ 12 และ 13 เป็นระยะเวลา 3 เดือน ทำการวัดอุณหภูมิ DO และ pH ข้อมูลที่ตรวจวัดได้ส่งผ่านมอดูลไร้สายซิกบีไปยังเครื่องแม่ข่าย และแต่ช่วงการทำงานของเซนเซอร์โนดมีการเก็บสถานะลงหน่วยความจำเพื่อใช้ในการวิเคราะห์ผล พบว่า ส่วนของ Simple-API สามารถทำงานได้ปกติ



รูปที่ 12 สถานที่ติดตั้งอุปกรณ์เซนเซอร์โนดร่วมกับ API ที่ออกแบบเพื่อทดสอบหาประสิทธิภาพ



รูปที่ 13 การติดตั้งเซนเซอร์โนดเพื่อทดสอบหาประสิทธิภาพ

4. การอภิปรายผลหรือการวิจารณ์และสรุป (Discussion and Conclusion)

การวิจัยครั้งนี้มีวัตถุประสงค์เพื่อออกแบบและเพิ่มความเป็นมิตรกับผู้ใช้งานในส่วนของการพัฒนาแอปพลิเคชันบนเซนเซอร์โนด ผลจากการวิจัยและพัฒนา ทำให้เกิด Simple-API จำนวน 4 มอดูล คือ มอดูลสำหรับการสื่อสาร มอดูลสำหรับการตรวจวัดค่าจากสิ่งแวดล้อม มอดูลสำหรับการประมวลผล และมอดูลสำหรับระบบพลังงานของเซนเซอร์โนด ผลทดสอบการใช้งานโลบาร์ทั้งหมดต่อพื้นที่หน่วยความจำพบว่า กรณิพัฒนาโปรแกรมประยุกต์โดยใช้หน่วยประมวลผลอาดูโน้ Mega2560 ทำการติดตั้งเฉพาะส่วนของ Simple-API ทั้งหมดใช้พื้นที่ในการพัฒนาโปรแกรม Sketch 2748 ไบต์ หรือคิดเป็นเพียงร้อยละ 1 และ ใช้พื้นที่หน่วยความจำ Global Variables น้อยกว่า 488 ไบต์ หรือคิดเป็นเพียงร้อยละ 5 ของหน่วยประมวลผล ด้านการใช้งานพบว่า API ที่สร้างขึ้น

การใช้งานโดยนำไปติดตั้งและพัฒนาส่วนโปรแกรมส่วนประยุกต์บน Arduino IDE มีความง่ายในการเรียกใช้งาน ซึ่งสามารถลดปัญหาเกี่ยวกับฮาร์ดแวร์และปัญหาในการโปรแกรมที่มีความซับซ้อน API ที่ออกแบบ มีการคืนค่าสถานะการทำงาน ดังนั้น จึงมั่นใจว่าข้อมูลที่รับในแต่ละคำสั่งนั้นสามารถนำไปใช้ประมวลผลตัดสินใจได้หรือไม่ ผลจากการออกแบบให้แต่ละ API มีส่วนของ watchdog timer ทำงานคู่ขนานไปนั้น ทำให้มั่นใจว่าการเรียกใช้ Simple-API จะไม่เกิดการทำงานค้าง

เบื้องต้นผล Simple-API ถูกนำไปขยายผลในงานวิจัยต่อยอดในการพัฒนาระบบระบบบริหารจัดการสัตว์น้ำอัจฉริยะ ทำให้สามารถลดเวลา งบประมาณและการพึ่งพา API ที่เป็นจากต่างประเทศ

การพัฒนางานวิจัยครั้งนี้มุ่งเน้นส่วนของโปรแกรมที่เรียกว่า API เป็นหลัก ยังขาดส่วนสำคัญๆ ที่ควรมีการศึกษาและวิจัย ได้แก่ 1) การพัฒนาฮาร์ดแวร์ให้สามารถทำงานร่วมกับ API ได้อย่างมีประสิทธิภาพ 2) การพัฒนา API ให้รองรับหน่วยประมวลผลตระกูลอื่นๆ และ 3) การพัฒนา API ให้รองรับเซนเซอร์หลากหลายประเภทหลากหลายผู้ผลิต เป็นต้น

5. กิตติกรรมประกาศ (Acknowledgements)

คณะผู้วิจัยขอขอบคุณมหาวิทยาลัยราชภัฏนครปฐม ที่สนับสนุนทุนวิจัย และขอขอบคุณคณะผู้ร่วมวิจัยทุกท่าน โดยเฉพาะอย่างยิ่ง คุณฐิติ สิทธิวงค์กุล ผู้ร่วมออกแบบและพัฒนา API ขอขอบคุณพี่ศิษยา มีศรีสุข เจ้าของฟาร์มดอนทอง เกษตรกรผู้เลี้ยงกิ้งก่ามรกต ในการให้ใช้สถานที่เพื่อเก็บข้อมูล

6. เอกสารอ้างอิง (References)

- [1] I. F. Akyildiz, T. Melodia and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *IEEE on Wireless Communications*, vol. 14, no. 6, pp. 32-39, Dec. 2007.
- [2] M. A. Razzaque, C. Bleakley and S. Dobson, "Compression in wireless sensor networks: A survey and comparative evaluation," *ACM Transactions on Sensor Network (TOSN)*, vol. 10, no. 1, pp. 537-568, Nov. 2013.
- [3] T. Sittivangkul, W. Cheunta, N. Chirdchoo, M. Saadi and L. Wuttisittikulkij, "Design and development of a wireless sensor mote prototype for laboratory usage", *ITC-CSCC 2014*, pp. 678-681, Jul., 2014.
- [4] M. A. Nasirudin, U. N. Za'bah, and O. Sidek, "Fresh water real-time monitoring system based on Wireless Sensor Network and GSM," in *Open Systems (ICOS)*, 2011 IEEE Conference on, 2011, pp. 354-357.
- [5] WISENSE Documentation [Intetnet]. 2014 [cited 2016 February 15] available from: <http://www.wisense.in/api/html/SystemOverview.html>
- [6] A. Grauballe, G. P. Perrucci, and F. H. P. Fitzek, "Opensensor - An open wireless sensor platform," in *4th International Mobile Multimedia Communications Conference*, 2008.
- [7] *PSU-Mote User Manual (REV. 1A ed.)* [Online]. Available: http://saturn.ee.psu.ac.th/~kittikhun/PSU_Mote/PSU-Mote_Manual.pdf
- [8] *Waspnote Document* [Online]. Available: <http://www.libelium.com/development/waspnote/documentation/waspnote-datasheet>