การพัฒนาต้นแบบโมตและส่วนต่อประสานโปรแกรมประยุกต์สำหรับงานเครือข่ายเซนเซอร์ไร้สาย

นายฐิติ สิทธิวางค์กูล

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ ที่ส่งผ่านทางบัณฑิตวิทยาลัย
The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository (CUIR)
are the thesis authors' files submitted through the University Graduate School.

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2558
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

A DEVELOPMENT OF A PROTOTYPE MOTE AND APPLICATION PROGRAMMING INTERFACE FOR WIRELESS SENSOR NETWORKS

Mr. Thiti Sittivangkul

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Electrical Engineering
Department of Electrical Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2015

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การพัฒนาต้นแบบโมตและส่วนต่อประสานโปรแกรมประยุกต์สำหรับงานเครือข่ายเซนเซอร์ไร้สาย
โดย	นายฐิติ สิทธิวางค์กูล
สาขาวิชา	วิศวกรรมไฟฟ้า
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.ลัญฉกร วุฒิสิทธิกุลกิจ
อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม	ดร.นิฏฐิตา เชิดชู

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์

(รองศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ สุวิทย์ นาคพีระยุทธ)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(รองศาสตราจารย์ ดร.ลัญฉกร วุฒิสิทธิกุลกิจ)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม

(ดร.นิฏฐิตา เชิดชู)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.พสุ แก้วปลั่ง)

.....กรรมการภายนอกมหาวิทยาลัย

(ดร.กมล เขมะรังษี)

ฐิติ สิทธิวางค์กูล : การพัฒนาต้นแบบโมตและส่วนต่อประสานโปรแกรมประยุกต์สำหรับงานเครือข่ายเซนเซอร์ไร้สาย (A DEVELOPMENT OF A PROTOTYPE MOTE AND APPLICATION PROGRAMMING INTERFACE FOR WIRELESS SENSOR NETWORKS) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: รศ. ดร.ลัญฉกร วุฒิสิทธิกุลกิจ, อ.ที่ปรึกษาวิทยานิพนธ์ร่วม: ดร.นิฏฐิตา เชิดชู, 110 หน้า.

วิทยานิพนธ์นี้นำเสนอการพัฒนาต้นแบบโนดเซนเซอร์ที่ไม่ใช้ระบบปฏิบัติการโดยอ้างอิงกับแพลตฟอร์มอาดูอิโน ต้นแบบที่พัฒนาขึ้นใช้หน่วยประมวลผลที่มีหน่วยความจำแฟลชขนาดใหญ่ 128 กิโลไบต์ แรมขนาด 16 กิโลไบต์ และติดตั้งวงจรที่จำเป็นสำหรับใช้งานเป็นโนดเซนเซอร์ ได้แก่ วงจรฐานเวลาจริง วงจรเพิ่มแรงดันแบตเตอรี่ วงจรตรวจวัดระดับแบตเตอรี่ ทั้งหมดรวมไว้บนแผงวงจรเดียวกัน นอกจากนี้ยังเพิ่มช่องต่อสำหรับเตรียมไว้รองรับแผงวงจรส่วนต่อขยาย อาทิ ชุดเซนเซอร์และเครื่องรับส่งสัญญาณวิทยุ ช่องต่อดังกล่าวสามารถเชื่อมต่อได้ทั้งสัญญาณแอนะล็อกและดิจิทัล การสื่อสารด้วยพอร์ตอนุกรม บัสเอสพีไอ และ บัส I^2C ส่วนต่อประสานโปรแกรมประยุกต์ที่นำเสนอมีชื่อเรียกว่า “Simple-API” ใช้รูปแบบการเขียนโปรแกรมเชิงวัตถุโดยแบ่งได้เป็น 4 ส่วนตามหน้าที่การทำงานของโนดเซนเซอร์ ได้แก่ ส่วนประมวลผล ส่วนตรวจวัดและสั่งการ ส่วนสื่อสารไร้สาย และ ส่วนควบคุมพลังงานและแบตเตอรี่ Simple-API จะทำหน้าที่ซ่อนการเข้าถึงเรจิสเตอร์ของอุปกรณ์และการเขียนโปรแกรมระดับล่าง จัดกลุ่มฟังก์ชันตามหน้าที่การทำงาน ป้องกันความผิดพลาดในขั้นตอนการพัฒนาและใช้งาน

ต้นแบบโนดเซนเซอร์และส่วนต่อประสานโปรแกรมประยุกต์ Simple-API ที่พัฒนาขึ้นเป็นประโยชน์ต่อการนำเครือข่ายเซนเซอร์มาประยุกต์ใช้งานจริงได้สะดวกมากขึ้นเพราะโครงสร้างที่ออกแบบให้ยืดหยุ่นตอบสนองต่อความต้องการได้ดี ค่าใช้จ่ายที่ลดลงเมื่อเทียบกับโนดเซนเซอร์ Waspmote ที่มีจำหน่ายในท้องตลาดและแพลตฟอร์มอาดูอิโน ที่สำคัญผู้ใช้งานสามารถพัฒนาโปรแกรมประยุกต์ได้โดยไม่ต้องศึกษาการทำงานของฮาร์ดแวร์ในระดับล่าง ตลอดจนรูปแบบการพัฒนาซอฟต์แวร์สั่งงานที่มีความซับซ้อนน้อยลงด้วย

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาควิชา	วิศวกรรมไฟฟ้า	ลายมือชื่อนิสิต	.....
สาขาวิชา	วิศวกรรมไฟฟ้า	ลายมือชื่อ อ.ที่ปรึกษาหลัก	.....
ปีการศึกษา	2558	ลายมือชื่อ อ.ที่ปรึกษาร่วม	.....

5570174921 : MAJOR ELECTRICAL ENGINEERING
KEYWORDS: WIRELESS / MICROCONTROLLER / HARDWARE DESIGN / APPLICATION PROGRAMMING INTERFACE / API

THITI SITTIVANGKUL: A DEVELOPMENT OF A PROTOTYPE MOTE AND APPLICATION PROGRAMMING INTERFACE FOR WIRELESS SENSOR NETWORKS. ADVISOR: ASSOC. PROF. LUNCHAKORN WUTTISITTIKULKIJ, Ph.D., CO-ADVISOR: NITTHITA CHIRDCHOO, Ph.D., 110 pp.

This dissertation proposes a development of the sensor node prototype and the application programming interface for wireless sensor networks (WSNs). The proposed sensor node is developed based on the Arduino platform on which requires no operating system to run. The node is also equipped with a processor with 128 KB of flash memory and 16 KB of RAM and various necessary circuits such as Real Time Clock, DC-DC Step up Converter and Battery Monitor. Moreover, we provide expansion socket interfaces on the board to be used with the radio and the sensor modules. These expansion sockets support both analog and digital interfaces include UART, SPI bus and I^2C bus. An application programming interface namely "Simple-API" is also purposed in this work. It includes four sets of function calls corresponding to the four subsystems of the sensor node architecture, namely, processing, sensing, communicating, and empowering subsystems. Simple-API provides a convenient way for the developers to create an application for WSNs, by sealing away the low-level programming, grouping function calls and preventing an error in the process of software development and its deployment.

The proposed platform allows application developers to focus more on the algorithm design and coding at the application level, thus providing a more convenient means to develop applications for WSNs. With the hardware and software of the sensor node we proposed in this work, we can achieve higher flexibility with lower complexity when comparing with both the Arduino and the Waspmote platform.

Department:	Electrical Engineering	Student's Signature .....
Field of Study:	Electrical Engineering	Advisor's Signature .....
Academic Year:	2015	Co-Advisor's Signature .....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ขอกราบขอบพระคุณอย่างสูงสำหรับความช่วยเหลืออย่างดียิ่งของ รองศาสตราจารย์ ดร.ลัญฉกร วุฒิสิทธิกุลกิจ อาจารย์ที่ปรึกษาวิทยานิพนธ์ และ ดร.นิฏฐิตา เชิดชู อาจารย์ที่ปรึกษาร่วม ซึ่งให้คำปรึกษารวมถึงข้อคิดเห็นต่าง ๆ ที่เป็นประโยชน์ต่อการวิจัยตลอดมาและขอกราบขอบพระคุณ อ.วีระศักดิ์ ชื่นตา สำหรับอุปกรณ์ที่ใช้ในการจัดทำฮาร์ดแวร์ ตลอดจนคำปรึกษาและแนวทางการแก้ปัญหาในงานวิจัยตลอดมา

สุดท้ายนี้ขอขอบคุณเพื่อน ๆ ในห้องวิจัยระบบสื่อสารและโทรคมนาคมที่ให้ความช่วยเหลือมาตลอดและกราบขอบพระคุณบิดามารดาซึ่งให้การสนับสนุนและกำลังใจมาโดยตลอดจนได้สำเร็จการศึกษา

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญ ..... ช

สารบัญตาราง ..... ฎ

สารบัญรูป ..... ฐ

บทที่ 1 บทนำ 1

1.1 ความสำคัญและที่มาของปัญหา 1

1.2 วัตถุประสงค์ 2

1.3 ขอบเขตงานวิจัย 3

1.4 ขั้นตอนและวิธีการดำเนินงาน 3

1.5 ประโยชน์ที่คาดว่าจะได้รับ 4

บทที่ 2 ความรู้พื้นฐานเครือข่ายเซนเซอร์ไร้สาย 5

2.1 ประวัติความเป็นมาของเครือข่ายเซนเซอร์ไร้สาย 5

2.2 องค์ประกอบและรูปแบบเครือข่ายเซนเซอร์ไร้สาย 6

2.3 โครงสร้างของโนดเซนเซอร์ 8

2.3.1 ส่วนประมวลผล 8

2.3.1.1 หน่วยประมวลผล 8

2.3.1.2 ช่องต่อประสาน 9

2.3.1.3 พื้นที่เก็บข้อมูล 12

2.3.2 ส่วนตรวจวัดและสั่งการ 12

2.3.2.1 เซนเซอร์ประเภทแอนะล็อก 12

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

หน้า

2.3.2.1 เซนเซอร์ประเภทดิจิทัล 12
2.3.3 ส่วนสื่อสารไร้สาย 15
2.3.4 ส่วนควบคุมพลังงานและแบตเตอรี่ 15
2.4 การพัฒนาโปรแกรมประยุกต์สำหรับโนดเซนเซอร์ 17
2.4.1 การพัฒนาโดยใช้ระบบปฏิบัติการ 17
2.4.2 การพัฒนาโดยไม่ใช้ระบบปฏิบัติการ 19
2.5 ตัวอย่างโนดเซนเซอร์ที่นำการใช้ในเครือข่ายเซนเซอร์ไร้สาย 20
2.5.1 โนดเซนเซอร์ที่พัฒนาขึ้นมาใช้งานโดยเฉพาะ 20
2.5.2 โนดเซนเซอร์ที่มีจำหน่ายทั่วไป 21
บทที่ 3 การพัฒนาต้นแบบโนดเซนเซอร์ 26
3.1 แพลตฟอร์มสำหรับการพัฒนาฮาร์ดแวร์ของโนดเซนเซอร์ 26
3.1.1 การพัฒนาฮาร์ดแวร์ที่ใช้ระบบปฏิบัติการ 26
3.1.1.1 ราสเบอรี พาย 27
3.1.1.2 บีเกิลโบน 30
3.1.2 การพัฒนาฮาร์ดแวร์ที่ไม่ใช้ระบบปฏิบัติการ 33
3.1.2.1 โนดเอ็มซียู 33
3.1.2.2 อาดูอิโน 35
3.2 แนวคิดการพัฒนาต้นแบบโนดเซนเซอร์ 37
3.3 โครงสร้างบอร์ดหลัก 39
3.3.1 ไมโครคอนโทรลเลอร์ 39
3.3.2 วงจรฐานเวลาจริง 40
3.3.3 ช่องต่อสำหรับแผงวงจรส่วนต่อขยาย 40
3.3.4 วงจรรวมแปลงระดับแรงดันไฟฟ้า 40

หน้า

3.3.5 วงจรรวมตรวจวัดแบตเตอรี่ 41
3.4 โครงสร้างบอร์ดส่วนต่อขยาย 41
3.4.1 อุปกรณ์เซนเซอร์ 42
3.4.1.1 เซนเซอร์ตรวจวัดค่าพีเอช 42
3.4.1.2 เซนเซอร์ตรวจวัดการละลายออกซิเจนในน้ำ 42
3.4.1.3 เซนเซอร์ตรวจวัดอุณหภูมิ 43
3.4.2 เครื่องรับส่งสัญญาณวิทยุ 44
3.4.2.1 เครื่องรับส่งสัญญาณวิทยุ Xbee PRO Zigbee S2 44
3.4.2.2 เครื่องรับส่งสัญญาณวิทยุ nRF24L01+ 45
3.4.2.3 เครื่องรับส่งสัญญาณวิทยุ HC-05 45
3.5 การเชื่อมต่อกับคอมพิวเตอร์ 46
3.5.1 การเชื่อมต่อโดยใช้เครื่องโปรแกรม 46
3.5.2 การเชื่อมต่อโดยใช้พอร์ตอนุกรม 46
3.6 แหล่งพลังงานของโนดเซนเซอร์ 47
3.6.1 ช่องยูเอสบี 48
3.6.2 แบตเตอรี่ 48
บทที่ 4 การพัฒนาส่วนต่อประสานโปรแกรมประยุกต์ 49
4.1 ภาพรวมการพัฒนาโปรแกรมประยุกต์สำหรับโนดเซนเซอร์ 49
4.2 การโปรแกรมลงบนโนดเซนเซอร์ 51
4.2.1 การโปรแกรมโดยใช้เครื่องโปรแกรม 51
4.2.2 การโปรแกรมโดยใช้เฟิร์มแวร์ 51
4.3 แนวคิดการพัฒนาส่วนต่อประสานโปรแกรมประยุกต์ 53
4.4 การออกแบบ Simple-API 54

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

หน้า

4.4.1 คลาส ProcessingSystemC 57
4.4.2 คลาส SensorAndActuatorC 60
4.4.3 คลาส WirelessCommunicationSystemC 62
4.4.4 คลาส PowerControlSystemC 65
บทที่ 5 ผลการดำเนินงาน 68
5.1 ต้นแบบโนดเซนเซอร์ 68
5.1.1 ลำดับการพัฒนาฮาร์ดแวร์ต้นแบบ 68
5.1.2 ข้อมูลด้านฮาร์ดแวร์ 72
5.2 ส่วนต่อประสานโปรแกรมประยุกต์ Simple-API 75
5.2.1 การสร้างและจัดกลุ่มฟังก์ชันการทำงาน 75
5.2.2 การลดจำนวนฟังก์ชันและยืดหยุ่นกับอุปกรณ์ 78
5.2.3 การป้องการความผิดพลาดการเรียกใช้งานและการแจ้งผลการทำงาน 81
5.3 การทดสอบการทำงานของต้นแบบโนดเซนเซอร์และ Simple-API 82
5.3.1 การสั่งงานอุปกรณ์บนโนดเซนเซอร์ 82
5.3.2 การส่งข้อมูลผ่านเครื่องรับส่งสัญญาณวิทยุ 84
บทที่ 6 สรุปผลและแนวทางการพัฒนาต่อในอนาคต 88
6.1 สรุปผลการดำเนินงาน 88
6.2 แนวทางพัฒนาต่อในอนาคต 89
รายการอ้างอิง 90
ภาคผนวก 95
ประวัติผู้เขียนวิทยานิพนธ์ 110

สารบัญตาราง

ตารางที่ 2.1 เซนเซอร์ที่นำไปใช้กับงานด้านต่าง ๆ 13
ตารางที่ 2.2 การเปรียบเทียบเซนเซอร์ประเภทแอนะล็อกและเซนเซอร์ประเภทดิจิทัล 14
ตารางที่ 2.3 คุณลักษณะของแบตเตอรี่ 16
ตารางที่ 3.1 รายละเอียดของบอร์ดราสเบอรี พาย Model A และ Model B 27
ตารางที่ 3.2 ระบบปฏิบัติการที่นำมาใช้กับบอร์ดราสเบอรี พาย 29
ตารางที่ 3.3 ลำดับการพัฒนาฮาร์ดแวร์ของโครงการบีเกิลโบน 31
ตารางที่ 3.4 รายละเอียดของบอร์ดอาดูอิโนแต่ละรุ่น 36
ตารางที่ 4.1 รายชื่อพารามิเตอร์ภายในมอดูล MODULE_DEVICE_INFO 57
ตารางที่ 4.2 รายชื่อพารามิเตอร์ภายในมอดูล MODULE_MCU 58
ตารางที่ 4.3 รายชื่อพารามิเตอร์ภายในมอดูล MODULE_REAL_TIME 58
ตารางที่ 4.4 รายชื่อพารามิเตอร์ภายในมอดูล MODULE_STATUS_LED 58
ตารางที่ 4.5 ฟังก์ชันสมาชิกแบบสาธารณะของคลาส ProcessingSystemC 59
ตารางที่ 4.6 รายชื่อพารามิเตอร์ภายในมอดูล MODULE_ATLAS_PH 60
ตารางที่ 4.7 รายชื่อพารามิเตอร์ภายในมอดูล MODULE_ATLAS_DO 61
ตารางที่ 4.8 รายชื่อพารามิเตอร์ภายในมอดูล MODULE_DALLAS_1820 61
ตารางที่ 4.9 ฟังก์ชันสมาชิกแบบสาธารณะของคลาส SensorAndActuatorC 62
ตารางที่ 4.10 รายชื่อพารามิเตอร์ภายในมอดูล MODULE_RADIO_24 63
ตารางที่ 4.11 รายชื่อพารามิเตอร์ภายในมอดูล MODULE_BLUETOOTH 63
ตารางที่ 4.12 รายชื่อพารามิเตอร์ภายในมอดูล MODULE_ZIGBEE 64
ตารางที่ 4.13 ฟังก์ชันสมาชิกแบบสาธารณะของคลาส WirelessCommunicationSystemC 64
ตารางที่ 4.14 รายชื่อพารามิเตอร์ภายในมอดูล MODULE_BATTERY 66
ตารางที่ 4.15 ฟังก์ชันสมาชิกแบบสาธารณะของคลาส PowerControlSystemC 67

ตารางที่ 5.1 ข้อมูลด้านเทคนิคต้นแบบโนดเซนเซอร์รุ่นที่ 2 73
ตารางที่ 5.2 การเปรียบเทียบข้อมูลด้านเทคนิคระหว่างต้นแบบโนดเซนเซอร์กับบอร์ดอาดูอิโน มาตรฐานและ Waspmote 74
ตารางที่ 5.3 รายชื่อฟังก์ชันสมาชิกแบบสาธารณะของแต่ละคลาส 76
ตารางที่ 5.4 การเปรียบเทียบคำสั่งที่ใช้ในการเริ่มทำงานของโนดเซนเซอร์ 77
ตารางที่ 5.5 ฟังก์ชันตั้งค่าอุปกรณ์ของ Simple-API เมื่อเทียบกับ Waspmote และอาดูอิโน 79
ตารางที่ 5.6 รูปแบบการเรียกใช้งานอุปกรณ์ของ Simple-API เมื่อเทียบกับ Waspmote และ อาดูอิโน 80
ตารางที่ 5.7 ผลการทำงานที่เป็นไปได้ของ Simple-API เมื่อเทียบกับ Waspmote และอาดูอิโน... 82

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญรูป

รูปที่ 2.1 ตัวอย่างการใช้งานเครือข่ายเซนเซอร์ไร้สาย 7
รูปที่ 2.2 โครงสร้างโนดเซนเซอร์ 8
รูปที่ 2.3 รูปแบบการเชื่อมต่อโดยใช้พอร์ตอนุกรม 10
รูปที่ 2.4 รูปแบบการเชื่อมต่ออุปกรณ์ด้วยบัสเอสพีไอ 11
รูปที่ 2.5 รูปแบบการเชื่อมต่ออุปกรณ์ด้วยบัส I^2C 11
รูปที่ 2.6 Mica 2 22
รูปที่ 2.7 IRIS Mote 22
รูปที่ 2.8 TelosB 22
รูปที่ 2.9 Tmote Sky 22
รูปที่ 2.10 WISENSE 24
รูปที่ 2.11 panStamp AVR 24
รูปที่ 2.12 Waspmote พร้อมเครื่องรับส่งสัญญาณวิทยุไร้สาย 25
รูปที่ 3.1 ราสเบอรี พาย Model B รุ่นที่ 3 28
รูปที่ 3.2 บอร์ด BeagleBone Black 30
รูปที่ 3.3 NodeMCU รุ่น 2 (ซ้าย) และ NodeMCU รุ่น 1 (ขวา) 34
รูปที่ 3.4 แผนผังแสดงลำดับการทำงานของกลุ่มคำสั่งภายในโปรแกรมประยุกต์ผู้ใช้ 36
รูปที่ 3.5 ภาพโครงสร้างแผงวงจรหลักและแผงวงจรส่วนขยายของต้นแบบโนดเซนเซอร์ 39
รูปที่ 3.6 เซนเซอร์ pH Circuit รุ่น 4.0 พร้อมหัววัด 42
รูปที่ 3.7 เซนเซอร์ D.O. Circuit รุ่น 6.0 พร้อมหัววัด 43
รูปที่ 3.8 เซนเซอร์ตรวจวัดอุณหภูมิ DS18B20 43
รูปที่ 3.9 เครื่องรับส่งสัญญาณวิทยุ Xbee PRO Zigbee S2 45
รูปที่ 3.10 เครื่องรับส่งสัญญาณวิทยุ nRF24L01+ 45

รูปที่ 3.11 เครื่องรับส่งสัญญาณวิทยุ HC-05 46
รูปที่ 3.12 การเชื่อมต่อระหว่างคอมพิวเตอร์และโนดเซนเซอร์ 47
รูปที่ 3.13 แหล่งพลังงานของโนดเซนเซอร์ 48
รูปที่ 4.1 แผนผังขั้นตอนการพัฒนาซอฟต์แวร์สำหรับโนดเซนเซอร์ 50
รูปที่ 4.2 แผนผังการทำงานของโปรแกรม Bootloader 52
รูปที่ 4.3 เครื่องโปรแกรม AVR ISP mkII 53
รูปที่ 4.4 ลำดับชั้นของการทำงานของ Simple-API 54
รูปที่ 4.5 ภาพรวมโครงสร้างการออกแบบพัฒนา Simple-API 55
รูปที่ 4.6 การเข้าถึงมอดูลและพารามิเตอร์ภายในคลาส 56
รูปที่ 4.7 โครงสร้างการออกแบบภายในคลาส ProcessingSystemC 57
รูปที่ 4.8 โครงสร้างการออกแบบภายในคลาส SensorAndActuatorC 60
รูปที่ 4.9 โครงสร้างการออกแบบภายในคลาส WirelessCommunicationSystemC 62
รูปที่ 4.10 โครงสร้างการออกแบบภายในคลาส PowerControlSystemC 65
รูปที่ 5.1 โครงสร้างภายในแผงวงจรหลักรุ่นที่ 1 68
รูปที่ 5.2 โครงสร้างช่องต่อแผงวงจรส่วนต่อขยายต้นแบบโนดเซนเซอร์รุ่นที่ 1 69
รูปที่ 5.3 แผงวงจรหลักรุ่นที่ 1 69
รูปที่ 5.4 โครงสร้างแผงวงจรของต้นแบบโนดเซนเซอร์หลักรุ่นที่ 2 ด้านหน้า 70
รูปที่ 5.5 โครงสร้างแผงวงจรของต้นแบบโนดเซนเซอร์หลักรุ่นที่ 2 ด้านหลัง 71
รูปที่ 5.6 แผงวงจรหลักรุ่นที่ 2 71
รูปที่ 5.7 โครงสร้างช่องต่อแผงวงจรส่วนต่อขยายต้นแบบโนดเซนเซอร์รุ่นที่ 2 72
รูปที่ 5.8 ข้อมูลอุณหภูมิเฉลี่ยของน้ำที่ตรวจวัดตลอดช่วงการทดลอง 83
รูปที่ 5.9 ข้อมูลแรงดันไฟฟ้าและความจุคงเหลือเฉลี่ยของแบตเตอรี่ตลอดช่วงการทดลอง 84
รูปที่ 5.10 ผลการทดสอบการส่งข้อมูลของเครื่องรับส่งสัญญาณวิทยุ nRFL01+ 85
รูปที่ 5.11 ผลการทดสอบการส่งข้อมูลของเครื่องรับส่งสัญญาณวิทยุ HC-05 86

Chulalongkorn University

รูปที่ 5.12 ผลการทดสอบการส่งข้อมูลของเครื่องรับส่งสัญญาณวิทยุ Xbee PRO S2..... 86

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของปัญหา

ในปัจจุบันการตรวจวัด การเฝ้าระวัง และการควบคุมมีความสำคัญมากในงานอุตสาหกรรม เกษตรกรรม สิ่งแวดล้อม การขนส่งสินค้า และการรักษาความปลอดภัย ความผิดพลาดเนื่องจากการขาดการตรวจวัด การเฝ้าระวัง และการจัดการอย่างเหมาะสมอาจก่อให้เกิดความเสียหายขึ้นกับงานสินค้านั้นได้ทั้งทางตรงและทางอ้อม อดีตจนถึงปัจจุบันงานดังกล่าวส่วนมากยังใช้คนเป็นหลัก ต้องอาศัยแรงงานจำนวนมากและอาจก่อให้เกิดความผิดพลาดจากการทำงานขึ้นได้ง่าย อีกทั้งยังใช้ค่าใช้จ่ายที่สูงในการปฏิบัติงาน ด้วยเหตุนี้จึงมีความพยายามนำระบบอัตโนมัติเข้ามาประยุกต์ใช้แทนการใช้แรงงานจากคน เพื่อเพิ่มประสิทธิภาพการทำงาน ลดความผิดพลาด และ สามารถช่วยลดต้นทุนในการปฏิบัติงานได้อีกด้วย

เทคโนโลยีเครือข่ายเซนเซอร์ (sensor network technology) เป็นระบบที่สามารถทำการตรวจวัด เฝ้าระวัง และควบคุมได้โดยอัตโนมัติ ไม่ต้องการใช้คนเพื่อปฏิบัติงานตลอดเวลา แรกเริ่มถูกนำมาใช้เพื่อเฝ้าระวังภัยทางการทหารในสนามรบ ภายหลังถูกพัฒนาต่อโดยได้รับความร่วมมือจากภาครัฐและเอกชนให้มีขนาดเล็กลง สื่อสารได้แบบไร้สาย ใช้พลังงานต่ำ สามารถติดตั้งเซนเซอร์บนแผงวงจรได้หลายประเภท และราคาอุปกรณ์ที่ลดลง ด้วยคุณสมบัติดังกล่าวทำให้เครือข่ายเซนเซอร์ไร้สายถูกเลือกและนำมาประยุกต์ใช้กับงานที่มีการตรวจวัดและเฝ้าระวังควบคุมทั้งด้านในหลาย ๆ ด้านได้ ดังนี้ ระบบตรวจวัดทางการแพทย์ระยะไกล ระบบดูแลผู้สูงอายุภายในที่พักอาศัย ระบบตรวจสอบควบคุมผลิตผลทางการเกษตร ระบบตรวจสอบและติดตามการขนส่งสินค้า ระบบควบคุมจราจรอัจฉริยะ และระบบการจัดการที่จอดรถ เป็นต้น

การนำเครือข่ายเซนเซอร์ไร้สาย (wireless sensor networks - WSN) มาประยุกต์ใช้งานยังอยู่ในวงจำกัดเฉพาะกลุ่มคนที่มีความรู้เฉพาะทาง บุคคลทั่วไปยังไม่สามารถนำเครือข่ายเซนเซอร์ไร้สายมาใช้ได้เนื่องจากข้อจำกัดในเรื่องฮาร์ดแวร์ (hardware) และ ซอฟต์แวร์ (software) ชุดอุปกรณ์สำเร็จรูปที่สามารถนำมาใช้งานมีตัวเลือกไม่มากนัก อุปกรณ์สำเร็จรูปบางชิ้นมีข้อจำกัดเรื่องการเชื่อมต่ออุปกรณ์อื่น ราคาจำหน่ายค่อนข้างสูง ต้องนำเข้ามาจากต่างประเทศซึ่งต้องผ่านขั้นตอนทางศุลกากรที่ยุ่งยาก ตลอดจนการซ่อมแซมหรือการรับประกันอุปกรณ์ที่ต้องส่งอุปกรณ์นั้นกลับไปยังประเทศผู้ผลิต ทางเลือกในการพัฒนาอุปกรณ์ขึ้นมาใช้ด้วยตัวเองต้องอาศัยความรู้ด้านอิเล็กทรอนิกส์ระบบสมองกลฝังตัว (embedded system) และเครื่องมือเฉพาะในการพัฒนา ส่วนของการพัฒนา

ซอฟต์แวร์สำหรับใช้งานเครือข่ายเซนเซอร์ไร้สายนั้นผู้พัฒนาจะต้องมีความรู้การใช้งานในระดับเรจิสเตอร์ (register) ของอุปกรณ์ ระบบการเชื่อมต่อระหว่างอุปกรณ์ การเขียนโปรแกรมขับอุปกรณ์ (device driver) ตลอดจนการเขียนโปรแกรมประยุกต์ของผู้ใช้ (user application) ผู้พัฒนาซอฟต์แวร์ส่วนใหญ่จะคุ้นชินกับการเขียนโปรแกรมในระดับโปรแกรมประยุกต์เท่านั้น การเปลี่ยนมาพัฒนาโปรแกรมสำหรับงานเครือข่ายเซนเซอร์ไร้สายทำให้ต้องใช้เวลาศึกษาวิธีการและเวลาที่ใช้พัฒนาเป็นเวลานาน ตลอดจนขั้นตอนตรวจสอบการทำงานที่อาจจะระบุหาสาเหตุของการทำงานที่ผิดพลาดจากการทำงานของฮาร์ดแวร์หรือซอฟต์แวร์

สำหรับงานวิจัยในวิทยานิพนธ์ฉบับนี้ นำเสนอทางเลือกสำหรับบุคคลที่สนใจนำเครือข่ายเซนเซอร์ไร้สายมาประยุกต์ใช้งานหรือศึกษาวิจัย โดยนำเสนอทั้งด้านฮาร์ดแวร์และซอฟต์แวร์ของโนดเซนเซอร์ ดังนี้ ด้านฮาร์ดแวร์นำเสนอต้นแบบโนดเซนเซอร์ที่พัฒนาอ้างอิงกับอาดูอิโน (Arduino) แยกเป็นบอร์ดหลักและบอร์ดต่อขยายเพื่อให้สามารถปรับเปลี่ยนเซนเซอร์และเครื่องรับส่งสัญญาณวิทยุได้ตามงานที่นำไปใช้ เพิ่มการเชื่อมต่อกับเครื่องส่งสัญญาณวิทยุที่ขาดหายไปของโนดเซนเซอร์ที่มีจำหน่าย ด้านซอฟต์แวร์นำเสนอส่วนต่อประสานโปรแกรมประยุกต์ (application programming interface - API) ให้ผู้พัฒนาเรียกใช้โดยจัดเป็นกลุ่มการทำงานเพื่อลดข้อความยุ่งยาก และความซับซ้อนในการพัฒนาโปรแกรมในระดับเรจิสเตอร์อุปกรณ์ โดยส่วนต่อประสานโปรแกรมประยุกต์จะถูกแบ่งออกเป็นกลุ่มตามหน้าที่การทำงานของโนด เพื่อทำให้การพัฒนาโปรแกรมประยุกต์ของผู้ใช้มีลักษณะความใกล้เคียงกับการพัฒนาโปรแกรมประยุกต์สำหรับคอมพิวเตอร์ทั่วไปที่นักพัฒนาส่วนใหญ่คุ้นเคย ตลอดจนลดขั้นตอนและเวลาในการนำเครือข่ายเซนเซอร์ไร้สายนำไปประยุกต์ใช้งานด้วย

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

1.2 วัตถุประสงค์

1. นำเสนอต้นแบบโนดเซนเซอร์ที่มีความยืดหยุ่นในการใช้งานเครือข่ายเซนเซอร์ไร้สายเพื่อแก้ไขข้อจำกัดของโนดเซนเซอร์ที่มีจำหน่าย

2. พัฒนาส่วนต่อประสานโปรแกรมประยุกต์เพื่ออำนวยความสะดวกกับผู้พัฒนาในการเรียกใช้งานฮาร์ดแวร์ของโนดเซนเซอร์และอุปกรณ์ที่เชื่อมต่อ เพื่อลดขั้นตอนการพัฒนาโปรแกรมประยุกต์สำหรับงานเครือข่ายเซนเซอร์ไร้สาย

1.3 ขอบเขตงานวิจัย

1. ออกแบบและพัฒนาต้นแบบโนดเซนเซอร์ แผงวงจรหลักเลือกใช้ไมโครคอนโทรลเลอร์ตระกูล AVR รุ่น 8 บิต ร่วมกับวงจรฐานเวลาจริง (real time clock - RTC) สามารถดาวน์โหลดโปรแกรมโนดผ่านพอร์ตอนุกรม และเครื่องโปรแกรม แหล่งพลังงานจากแบตเตอรี่หรือพอร์ตยูเอสบี สามารถติดเชื่อมต่ออุปกรณ์ภายนอกได้ดังนี้

1.1 ระบบตรวจวัดสามารถเชื่อมต่อเซนเซอร์ประเภทแอนะล็อกและดิจิทัล

1.2 ระบบสื่อสารไร้สายสามารถเชื่อมต่อเครื่องรับส่งสัญญาณวิทยุ nRF24L01+ เครื่องรับส่งสัญญาณวิทยุ XbeePRO S2 Zigbee และ เครื่องรับส่งสัญญาณวิทยุบูลทูท (bluetooth)

2. ออกแบบและพัฒนาส่วนต่อประสานโปรแกรมประยุกต์สำหรับเรียกใช้งานอุปกรณ์บนโนดเซนเซอร์โดยใช้ภาษาโปรแกรมภาษาซี++ โดยแบ่งเป็น 4 ส่วนหลักดังนี้

2.1 ส่วนประมวลผล ทำหน้าที่ควบคุมการทำงานของไมโครคอนโทรลเลอร์หลัก ระบบฐานเวลาจริง การตั้งเวลาปลุก การควบคุมการทำงานของหลอดแอลอีดี (LED)

2.2 ส่วนสื่อสารไร้สาย ทำหน้าที่สร้างการเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์และเครื่องรับส่งสัญญาณวิทยุ การปรับตั้งค่าพารามิเตอร์ที่สำคัญของเครื่องรับส่งสัญญาณวิทยุ และการรับส่งข้อมูล

2.3 ส่วนตรวจวัดและสั่งการ ทำหน้าที่สร้างการเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์และเซนเซอร์ การปรับตั้งค่าพารามิเตอร์ที่ใช้สำหรับตรวจวัด และการอ่านค่าจากเซนเซอร์

2.4 ส่วนควบคุมพลังงานและแบตเตอรี่ ทำหน้าที่ตรวจวัดพลังงานที่เหลือในแบตเตอรี่ ควบคุมการใช้งานวงจรพิเศษภายในของไมโครคอนโทรลเลอร์

1.4 ขั้นตอนและวิธีการดำเนินงาน

1. ศึกษางานวิจัยการนำเครือข่ายเซนเซอร์ไร้สายประยุกต์ใช้งาน

2. พิจารณาโนดเซนเซอร์ที่มีการนำมาใช้ในงานวิจัยและที่มีจำหน่ายในปัจจุบัน

3. พิจารณาขั้นตอนและวิธีในการพัฒนาโปรแกรมประยุกต์สำหรับงานเครือข่ายเซนเซอร์ไร้สาย

4. นำเสนอต้นแบบโนดเซนเซอร์และส่วนต่อประสานโปรแกรมประยุกต์ที่สร้างขึ้น

5. ทดสอบการทำงานของต้นแบบโนดเซนเซอร์และการเรียกใช้ส่วนต่อประสานโปรแกรมประยุกต์

6. เขียนบทความทางวิชาการและนำเสนอผลงาน

7. สรุปผลการทำวิจัยและจัดทำเล่มวิทยานิพนธ์ฉบับสมบูรณ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ต้นแบบโนดเซนเซอร์ที่มีความยืดหยุ่นในการใช้งานสำหรับการใช้งานจริงและงานด้านการวิจัย

2. ลดขั้นตอนในการพัฒนาทั้งในส่วนฮาร์ดแวร์และซอฟต์แวร์ของการประยุกต์ใช้งานเครือข่ายเซนเซอร์ไร้สาย

3. เพิ่มทางเลือกในการนำเครือข่ายเซนเซอร์ไร้สายมาประยุกต์ใช้งานในงานอุตสาหกรรม เกษตรกรรม การตรวจวัดสิ่งแวดล้อม และงานด้านอื่น ๆ สำหรับผู้ใช้งานในประเทศมากยิ่งขึ้น

4. เพื่อเป็นแนวทางส่งเสริมการนำงานเครือข่ายเซนเซอร์ไร้สายมาประยุกต์ใช้งานในประเทศมากขึ้น

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 2
ความรู้พื้นฐานเครือข่ายเซนเซอร์ไร้สาย

2.1 ประวัติความเป็นมาของเครือข่ายเซนเซอร์ไร้สาย

งานวิจัยเครือข่ายเซนเซอร์ไร้สายมีต้นกำเนิดโดยย้อนกลับไปยังโครงการเครือข่ายเซนเซอร์แบบกระจาย (Distributed Sensor Networks – DSNs) ในปี ค.ศ. 1980 ของหน่วยงานสำนักงานโครงการวิจัยขั้นสูงของกระทรวงกลาโหมสหรัฐอเมริกา (Defense Advance Research Project Agency – DARPA) คาดการว่าโครงการดังกล่าวมีการติดตั้งโนดเซนเซอร์จำนวนมาก สามารถทำงานได้เองโดยอัตโนมัติและสื่อสารกันได้โดยสามารถเลือกใช้เส้นทางที่เหมาะสมที่สุดผ่านโนดต่าง ๆ ภายในเครือข่าย ในเวลาต่อมานักวิจัยที่มหาวิทยาลัยคาร์เนกีเมลลอน (Carnegie Mellon University) ได้พัฒนาระบบปฏิบัติการที่มีชื่อเรียกว่า แอกเซนท์ (Accent) มีความยืดหยุ่นในการใช้งาน อนุญาตให้เข้าถึงทรัพยากรระบบของเครือข่ายเซนเซอร์แบบกระจาย สามารถทำงานต่อได้แม้มีการผิดพลาดขึ้น ตัวอย่างการนำโครงการเครือข่ายเซนเซอร์แบบกระจายไปประยุกต์ใช้งานได้แก่ ระบบติดตามเฮลิคอปเตอร์โดยใช้คลื่นเสียง โดยติดตั้งไมโครโฟนแบบกระจายร่วมกับเทคนิคในการเปรียบเทียบเสียงโดยสถาบันเทคโนโลยีแมสซาชูเซตส์ (Massachusetts Institute of Technology) ประเทศสหรัฐอเมริกา ขณะนั้นนักวิจัยมีความหวังที่จะนำวิสัยทัศน์ของโครงการเครือข่ายเซนเซอร์แบบกระจายมาประยุกต์ใช้ แต่เทคโนโลยีในขณะนั้นยังไม่พร้อมให้นำใช้งานโดยเฉพาะอย่างยิ่งเซนเซอร์ที่มีขนาดใหญ่ (ขนาดประมาณกล่องใส่รองเท้าหรือใหญ่กว่า) ซึ่งเป็นข้อจำกัดในการนำไปงานประยุกต์ อีกทั้งโครงการเครือข่ายเซนเซอร์แบบกระจายนี้ยังไม่ได้ให้ความสำคัญกับการสื่อสารแบบไร้สายมากนัก ต่อมาในปี ค.ศ. 1998 ได้พบแนวทางที่จะพัฒนาเครือข่ายเซนเซอร์ไร้สายขึ้นจากความก้าวหน้าของการสร้างหน่วยประมวลผล การสื่อสารไร้สาย และเซนเซอร์ขนาดเล็กลง [1]

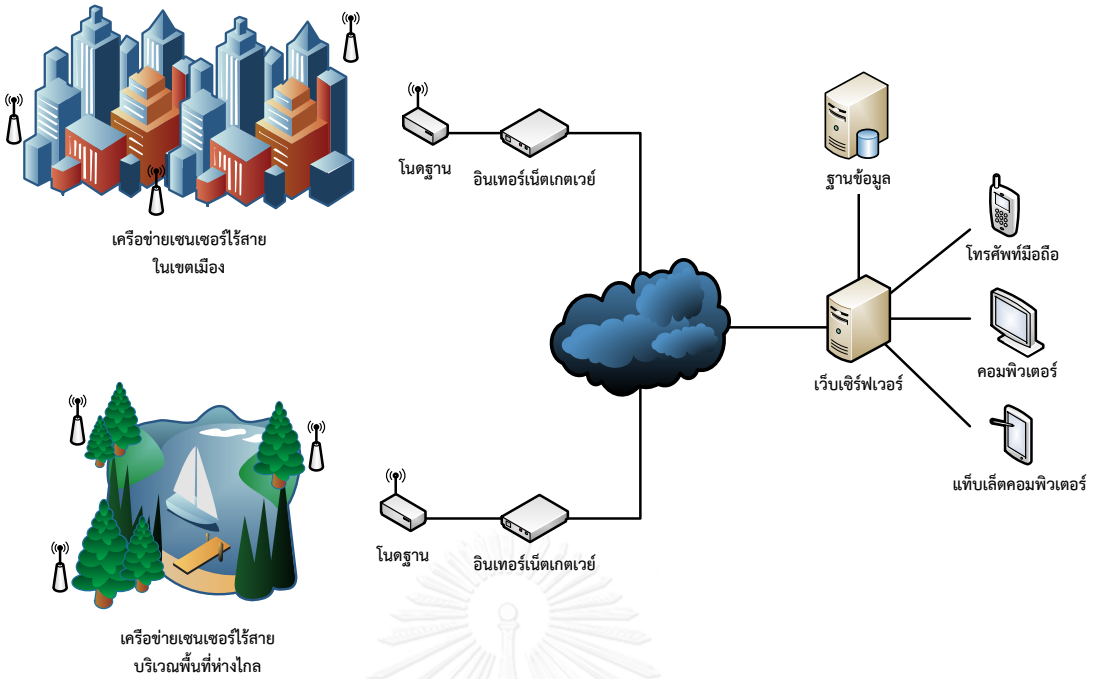
การวิจัยเครือข่ายเซนเซอร์ไร้สายในปัจจุบัน [2] จึงมุ่งเน้นไปยังการพัฒนาอุปกรณ์ให้มีขนาดเล็กลง ราคาลดลงจากเดิม รูปแบบเครือข่ายที่สามารถปรับเปลี่ยนได้ตามสภาพแวดล้อมที่นำไปใช้งาน นอกจากนี้ยังมีการพัฒนาให้มีความสามารถมากขึ้น เช่น การพัฒนาการใช้งานเครือข่ายแบบเฉพาะกิจ (Ad-hoc network) การทำงานหลายงานกันในเวลาเดียวกัน (multitasking) และ ความสามารถการโปรแกรมอุปกรณ์ซ้ำใหม่ได้ เป็นต้น ในเวลาต่อมาสถาบันวิศวกรไฟฟ้าและอิเล็กทรอนิกส์ (Institute of Electrical and Electronic Engineers - IEEE) ได้ออกข้อกำหนดมาตรฐานสำหรับเครือข่ายไร้สายพื้นที่ส่วนตัวที่มีอัตราการส่งข้อมูลต่ำ IEEE 802.15.4 จากมาตรฐานดังกล่าวทำให้กลุ่มซิกบี (ZigBee Alliance) นำเสนอมาตรฐานซิกบี (Zigbee) ซึ่งเป็นโพรโทคอล (protocol) การสื่อสารระดับสูงของการสื่อสารไร้สายในเครือข่ายเซนเซอร์ไร้สาย เน้นการใช้งานในบ้านเรือน สำนักงาน และ

โรงงานอุตสาหกรรม ปัจจุบันเครือข่ายเซนเซอร์ไร้สายถูกนำมาประยุกต์ใช้งานอย่างกว้างขวางในงานด้านต่าง ๆ [3-8] เช่น การเกษตรกรรม การแพทย์สาธารณสุข การตรวจวัดในงานอุตสาหกรรม งานเฝ้าระวังภัยพิบัติทางธรรมชาติ งานจราจรอัจฉริยะ และ งานติดตามและระบบพิกัดของสิ่งของ

สำหรับประเทศไทยมีงานวิจัยเครือข่ายเซนเซอร์ไร้สายนำไปประยุกต์ใช้งาน ได้แก่ การเพิ่มประสิทธิภาพการเพาะเลี้ยงกุ้ง [9] การจัดการพลังงานที่ใช้ภายในบ้านและสำนักงาน [10, 11] ระบบเฝ้าระวังและช่วยดูแลผู้สูงอายุภายในบ้าน [12] อุปกรณ์แบบสวมใส่สำหรับตรวจวัดสัญญาณชีพ [13] การใช้งานในห้องพักผู้ป่วยเพื่อตรวจวัดทางการแพทย์ [14] เทคโนโลยีเครือข่ายเซนเซอร์ไร้สายจึงกลายเป็นเทคโนโลยีที่สำคัญอย่างหนึ่งในศตวรรษนี้ สำหรับประเทศไทยมีการบรรจุนโยบายการประยุกต์ใช้งานเครือข่ายเซนเซอร์สำหรับงานด้านการเกษตรเพื่อเพิ่มผลผลิต การตรวจสอบเฝ้าระวังสิ่งแวดล้อมของภาคอุตสาหกรรม และการเฝ้าระวังภัยพิบัติทางธรรมชาติ ในกรอบนโยบายเทคโนโลยีสารสนเทศและการสื่อสารระยะ พ.ศ. 2554-2563 [15] และในหลายประเทศรวมทั้งประเทศจีนได้บรรจุหัวข้อเครือข่ายเซนเซอร์เข้าไปในแผนการวิจัยของชาติอีกด้วย [1]

2.2 องค์ประกอบและรูปแบบเครือข่ายเซนเซอร์ไร้สาย

เครือข่ายเซนเซอร์ไร้สายประกอบด้วยอุปกรณ์ขนาดเล็กเรียกว่า โนดเซนเซอร์ (sensor node) หรือ โมต (motes) [3] ทำหน้าที่ ตรวจวัดข้อมูลที่สนใจ ประมวลผลข้อมูล ตัดสินใจจากข้อมูลที่ตรวจวัด สื่อสารข้อมูลภายในเครือข่าย และ อาจทำหน้าที่สั่งงานได้อีกด้วย โนดเซนเซอร์ประกอบด้วยหน่วยประมวลผล เครื่องรับส่งสัญญาณวิทยุ หน่วยความจำ วงจรแปลงสัญญาณเซนเซอร์ และแหล่งจ่ายพลังงาน ด้วยข้อจำกัดเรื่องทรัพยากรที่ติดตั้งในโนดเซนเซอร์รูปแบบการทำงานของโนดเซนเซอร์จึงมีลักษณะเฉพาะ เช่น การตรวจวัดข้อมูลจะทำตามรอบเวลาที่ตั้งไว้ ข้อมูลที่วัดได้ต้องไม่ต้องการประมวลผลที่ซับซ้อน อัตราการส่งข้อมูลต่ำ (ประมาณ 10 – 100 กิโลบิตต่อวินาที) และระยะทางการสื่อสารระหว่างโนดไม่ไกล (น้อยกว่า 100 เมตร) การสื่อสารข้ามระหว่างเครือข่าย หรือ การส่งข้อมูลกลับมายังระบบส่วนกลางทำโดยผ่านโนดเซนเซอร์ที่ทำหน้าที่เป็นโนดฐาน (base node) และติดตั้งอินเทอร์เน็ตเกตเวย์ (internet gateway) เพื่อเชื่อมต่อกับเซิร์ฟเวอร์ (server) ผ่านเครือข่ายอินเทอร์เน็ต ข้อมูลที่ส่งกลับมานั้นจะส่งไปเก็บลงในฐานข้อมูลเพื่อนำไปวิเคราะห์ผล และแสดงผลต่อไป โครงสร้างตัวอย่างการใช้งานเครือข่ายเซนเซอร์ไร้สายในบริเวณพื้นที่ต่าง ๆ โดยมีการส่งข้อมูลผ่านเครือข่ายอินเทอร์เน็ตแสดงในรูปที่ 2.1

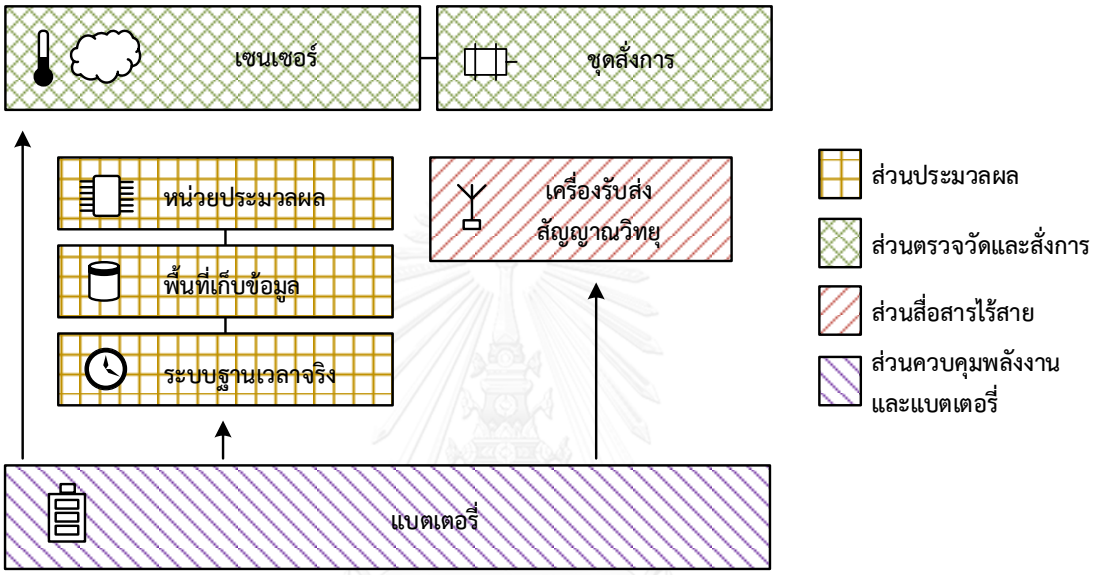


รูปที่ 2.1 ตัวอย่างการใช้งานเครือข่ายเซนเซอร์ไร้สาย

การนำเครือข่ายเซนเซอร์ไปประยุกต์ใช้งาน โนดเซนเซอร์อาจมีรูปแบบการติดตั้งกระจายลงในพื้นที่ใช้งานหรือติดตั้งตามตำแหน่งที่มีการวางแผนไว้ล่วงหน้า วิธีการติดตั้งทั้งสองมีจุดเด่นและข้อสังเกตดังนี้ การติดตั้งแบบกระจายลงในพื้นที่ใช้งานมีความสะดวกสำหรับงานในพื้นที่ขนาดใหญ่ ใช้โนดเซนเซอร์จำนวนมาก แต่การบำรุงรักษาทำให้โนดเซนเซอร์ทำงานได้ปกติทำได้ยาก ค่าใช้จ่ายสูงเนื่องจากจำนวนของโนดที่ใช้งานเป็นจำนวนมาก การติดตั้งแบบการวางแผนไว้ล่วงหน้ามีจุดเด่นเรื่องพื้นที่การทำงานที่ครอบคลุมได้ตามความต้องการ ใช้จำนวนโนดเซนเซอร์ไม่มาก ค่าใช้จ่ายในการติดตั้งและซ่อมบำรุงต่ำกว่าการติดตั้งแบบกระจายลงในพื้นที่ รูปแบบการติดตั้งจึงเป็นปัจจัยหนึ่งที่ส่งผลต่อประสิทธิภาพการรับส่งข้อมูลซึ่งส่งผลถึงประสิทธิภาพการทำงานรวมของเครือข่าย เนื่องจากการสื่อสารเป็นกระบวนการที่ใช้พลังงานสูง รูปแบบการรับส่งข้อมูลจึงต้องนำเทคนิคเข้ามาช่วยจัดการสื่อสารให้มีประสิทธิภาพสูงสุด เพื่อให้โนดเซนเซอร์สามารถทำงานได้ตามระยะเวลาและจุดประสงค์ของงานที่นำไปประยุกต์ใช้ โดยไม่ต้องเข้าไปทำการบำรุงรักษาระหว่างที่ใช้งาน

2.3 โครงสร้างของโนดเซนเซอร์

โนดเซนเซอร์เป็นอุปกรณ์อิเล็กทรอนิกส์ขนาดเล็กประกอบด้วยหน่วยประมวลผล หน่วยความจำ อุปกรณ์ตรวจวัด เครื่องรับส่งสัญญาณวิทยุ และ แหล่งพลังงาน การพัฒนาฮาร์ดแวร์ชองโนดเซนเซอร์มีรูปแบบคล้ายกับอุปกรณ์ที่ใช้งานระบบสมองกลฝังตัว การพัฒนาสามารถแบ่งตามหน้าที่การทำงานของโนดออกเป็นส่วนย่อย ๆ ดังแสดงแสดงในรูปที่ 2.2 โดยมีรายละเอียดส่วนการทำงานย่อยทั้ง 4 ส่วนดังนี้



รูปที่ 2.2 โครงสร้างโนดเซนเซอร์

2.3.1 ส่วนประมวลผล

ส่วนประมวลผลเป็นระบบกลางของการทำงานภายในโนดเซนเซอร์ เชื่อมต่อกับส่วนการทำงานอื่น ทำหน้าที่หลักในการควบคุมสั่งงานอุปกรณ์ทั้งหมดที่ติดตั้งบนโนดเซนเซอร์ ประมวลผลข้อมูลและตัดสินใจทำงานจากข้อมูลที่ตรวจวัดได้ ภายในส่วนประมวลผลยังครอบคลุมถึงระบบฐานเวลาจริงและการเก็บบันทึกข้อมูลลงในหน่วยความจำอีกด้วย เราสามารถแยกส่วนประกอบของส่วนประมวลผลได้เป็น 4 ส่วนดังนี้

2.3.1.1 หน่วยประมวลผล

หน่วยประมวลผล (processing unit) เปรียบเสมือนสมองของระบบสร้างขึ้นเป็นวงจรรวม (integrated circuit - IC) ประกอบด้วยหน่วยประมวลผล (processor) ช่องทางเข้า/ออก (input/output port) และ อาจรวมหน่วยความจำ (memory) ประเภทแฟลชไว้ในวงจรรวมเดียวกัน นอกจากนี้อาจมีการเพิ่มวงจรทำหน้าพิเศษอื่น ๆ เช่น วงจรเวลาหรือวงจรนับสัญญาณ

(timer/counter) วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล (analog-to-digital converter) อีกด้วย

ปัจจุบันหน่วยประมวลผลมีให้เลือกหลายประเภทได้แก่ หน่วยประมวลผลสัญญาณดิจิทัล (digital signal processing - DSP) วงจรรวมที่ทำหน้าที่เฉพาะ (application-specific integrated circuit) วงจรรวมเอฟพีจีเอ (field programming gate arrays - FPGA) และ ไมโครคอนโทรลเลอร์ (microcontroller) การเลือกใช้หน่วยประมวลผลต้องพิจารณาถึงประสิทธิภาพการใช้งานรวมถึงการพลังงานที่ใช้ของหน่วยประมวลผลนั้นด้วย ยกตัวอย่างเช่น โนดเซนเซอร์ที่ต้องการประมวลผลภาพถ่ายหรือสัญญาณวีดิโอต้องใช้หน่วยประมวลผลที่มีประสิทธิภาพสูงซึ่งต้องการใช้พลังงานที่สูงตามไปด้วย โนดเซนเซอร์ที่ใช้งานภายนอกอาคารหรือพื้นที่ที่เข้าถึงได้ยากอาจต้องเลือกหน่วยประมวลผลที่ใช้พลังงานต่ำเพื่อให้โนดเซนเซอร์สามารถใช้งานได้ยาวนานแต่ประสิทธิภาพการประมวลผลก็จะลดลงด้วยเช่นกัน

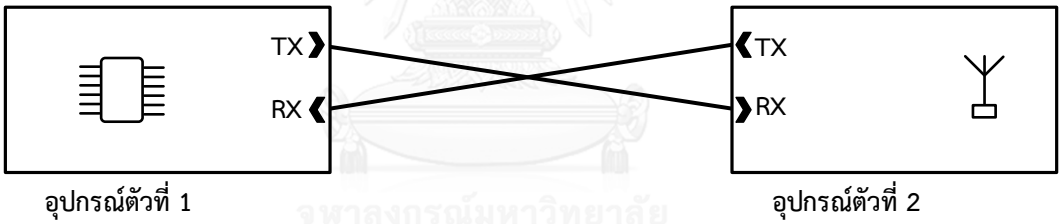
โนดเซนเซอร์ในปัจจุบันมีแนวโน้มเลือกไมโครคอนโทรลเลอร์มาใช้งานเป็นหน่วยประมวลผลมากขึ้น [16] เนื่องจากในปัจจุบันไมโครคอนโทรลเลอร์ถูกพัฒนาให้มีประสิทธิภาพสูงมากขึ้น มีการสร้างวงจรหน้าที่พิเศษกล่าวมาข้างต้นรวมไว้ในอุปกรณ์เพียงชิ้นเดียว หน่วยความจำภายในมีขนาดให้เลือกใช้ได้หลากหลาย ใช้พลังงานต่ำขณะทำงานในโหมดนอนหลับ (sleep mode) และราคาจำหน่ายต่ำลง ทำให้ต้นทุนการสร้างโนดเซนเซอร์ลดลงตามไปด้วย อีกทั้งมีความยืดหยุ่นในขั้นตอนการพัฒนาและการใช้งานจริงมากกว่าหน่วยประมวลผลประเภทอื่น ๆ อีกด้วย

2.3.1.2 ช่องต่อประสาน

ช่องต่อประสาน (interface) ทำหน้าที่เป็นช่องทางเข้าหรือออกของสัญญาณจากหน่วยประมวลผลเชื่อมต่อไปยังอุปกรณ์ภายนอก สามารถแบ่งตามลักษณะสัญญาณได้ 2 ประเภท ได้แก่ สัญญาณดิจิทัลและสัญญาณแอนะล็อก การต่อประสานกับหน่วยประมวลผลจะใช้สัญญาณดิจิทัลที่มีระดับแรงดันขนาด 5 โวลต์ หรือ 3.3 โวลต์ แทนสัญญาณดิจิทัลมีค่าเท่ากับ 1 และ สัญญาณที่มีขนาดแรงดัน 0 – 0.7 โวลต์ แทนสัญญาณดิจิทัลมีค่าเท่ากับ 0 สัญญาณดิจิทัลดังกล่าวอาจได้มาจากเซนเซอร์แม่เหล็ก สัญญาณปลุกจากวงจรนาฬิกา หรือ สัญญาณเตือนจากวงจรตรวจวัดพลังงานในแบตเตอรี่เมื่อระดับพลังงานในแบตเตอรี่ต่ำ หน่วยประมวลผลอาจมีวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลภายในเพื่อแปลงระดับสัญญาณแอนะล็อกให้ออกมาเป็นข้อมูลเพื่อนำไปประมวลผลต่อได้ โดยความละเอียดในการแปลงสัญญาณแอนะล็อกสามารถพิจารณาได้จากจำนวนของบิตที่ใช้ในการแทนระดับสัญญาณแอนะล็อก ถ้าจำนวนบิตที่ใช้มีจำนวนมากข้อมูลที่ได้จากการแปลงสัญญาณแอนะล็อกก็จะมีความละเอียดสูง สัญญาณประเภทแอนะล็อกจะพบได้ในเซนเซอร์

ตรวจวัดอุณหภูมิ น้ำหนัก ความต้านทานทางไฟฟ้า เป็นต้น นอกจากช่องต่อประสานสัญญาณดิจิทัล และช่องต่อประสานสัญญาณแอนะล็อกแล้วยังมีช่องต่อประสานเฉพาะอื่น ๆ อีกที่นำมาใช้เชื่อมต่ออุปกรณ์ภายในโนดเซนเซอร์ได้แก่

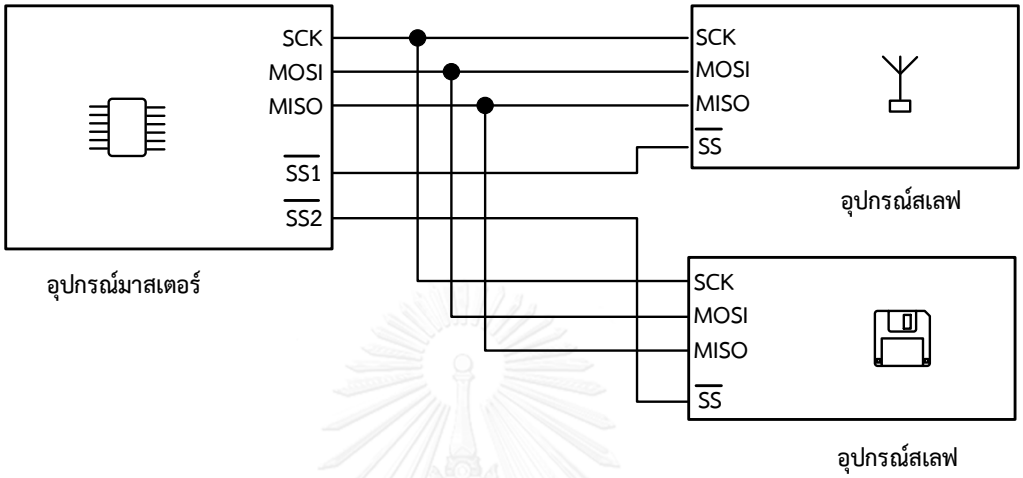
พอร์ตอนุกรม (serial port) เป็นช่องต่อประสานที่ใช้สัญญาณดิจิทัล รูปแบบการรับส่งข้อมูลเป็นแบบอนุกรมสามารถสื่อสารได้สองทางในเวลาเดียวกัน (full duplex) สามารถใช้สายสัญญาณเพียง 2 เส้น คือ TX สำหรับส่งข้อมูลออก และ RX สำหรับส่งข้อมูลเข้า การต่อสายสัญญาณจะต้องสลับสายสัญญาณ RX และ TX ของอุปกรณ์ทั้งสองเพื่อให้สามารถสื่อสารระหว่างกันได้ดังแสดงในรูปที่ 2.3 การสื่อสารผ่านพอร์ตอนุกรมจะเป็นการสื่อสารระหว่างอุปกรณ์ 2 ตัว เท่านั้นไม่สามารถเชื่อมต่ออุปกรณ์ให้เป็นบัสข้อมูลได้ ไมโครคอนโทรลเลอร์จะมีช่องต่อประสานประเภทดังกล่าวอย่างน้อย 1 ช่อง นิยมใช้เป็นช่องทางสำหรับสื่อสารกับคอมพิวเตอร์ นอกจากนี้ยังอาจใช้เชื่อมต่อกับโมเด็ม (modem) เพื่อเชื่อมต่อกับอินเทอร์เน็ต ในปัจจุบันมีการติดตั้งวงจรแปลงสัญญาณจากพอร์ตอนุกรมให้สามารถเชื่อมต่อกับคอมพิวเตอร์ผ่านพอร์ตยูเอสบี (USB port) ได้เพื่อใช้งานกับคอมพิวเตอร์รุ่นใหม่ที่ไม่มีช่องเชื่อมต่ออนุกรม



รูปที่ 2.3 รูปแบบการเชื่อมต่อโดยใช้พอร์ตอนุกรม

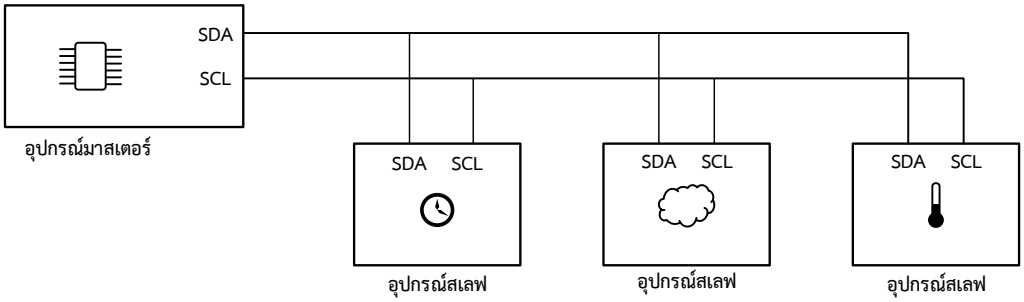
บัสเอสพีไอ (serial peripheral interface bus - SPI) เป็นช่องต่อประสานที่ใช้สัญญาณดิจิทัล รูปแบบการรับส่งข้อมูลภายในบัสเป็นแบบอนุกรม ใช้สายสัญญาณทั้งหมด 4 เส้น คือ MISO (master-in/slave-out) MOSI (master-out/slave-in) SCK (serial clock) และ CS (chip select) การสื่อสารระหว่างอุปกรณ์จะมีการทำงานในลักษณะมาสเตอร์และสเลฟ (master – slave) อุปกรณ์ที่ต้องการเริ่มสื่อสารจะทำหน้าที่มาสเตอร์ ส่วนอุปกรณ์ที่สื่อสารด้วยจะทำหน้าที่เป็นสเลฟ โดยการทำงานอุปกรณ์ที่ทำหน้าที่มาสเตอร์จะส่งข้อมูลไปบอกอุปกรณ์ที่จะสื่อสารด้วยผ่านสายสัญญาณ CS จากนั้นอุปกรณ์มาสเตอร์จะส่งสัญญาณนาฬิกาผ่านสายสัญญาณ CSK พร้อมกับข้อมูล MOSI ทีละบิตตามจังหวะสัญญาณนาฬิกาไปให้อุปกรณ์ที่สื่อสารด้วย เมื่อรับสัญญาณนาฬิกาแล้วอุปกรณ์สเลฟจะส่งข้อมูลออกมาทีละบิตผ่านสายสัญญาณ MISO เช่นกัน ซึ่งคล้ายกับการทำงานของชิฟต์

เรจิสเตอร์ (shift register) การใช้งานบัสเอสพีไอในโนดเซนเซอร์นั้น ไมโครคอนโทรลเลอร์จะทำหน้าที่เป็นอุปกรณ์มาสเตอร์เพื่อขออ่านหรือส่งข้อมูลไปยังอุปกรณ์สเลฟที่นำมาเชื่อมต่อ เช่น เครื่องรับส่งสัญญาณวิทยุ ไมโครเอสดีการ์ด และ วงจรรวมของหน่วยความจำภายนอก



รูปที่ 2.4 รูปแบบการเชื่อมต่ออุปกรณ์ด้วยบัสเอสพีไอ

บัส I^2C (Inter Integrate Circuit Bus – I^2C) เป็นช่องต่อประสานที่ใช้สัญญาณดิจิทัล รูปแบบการรับส่งข้อมูลภายในบัสเป็นการรับส่งข้อมูลแบบอนุกรม พัฒนาขึ้นโดยบริษัท ฟิลลิป เซมิคอนดักเตอร์ (Philips Semiconductor) ใช้สายสัญญาณเพียง 2 เส้น คือ SDA (serial data analyzer) และ SCK (serial clock) อุปกรณ์ที่นำมาเชื่อมต่อภายในบัสจะถูกกำหนดหมายเลขประจำตัวที่ไม่ซ้ำกันจากผู้ผลิต การสื่อสารกับอุปกรณ์ภายในบัสจะต้องส่งหมายเลขประจำตัวของอุปกรณ์ที่ต้องการสื่อสารก่อนการสื่อสารจริงจะเริ่มขึ้น บัสสามารถเชื่อมต่ออุปกรณ์ได้มากและใช้สายสัญญาณที่น้อยกว่าการใช้บัสเอสพีไอ การเชื่อมต่อด้วยบัส I^2C นิยมใช้กับอุปกรณ์ เช่น เซนเซอร์วัดอุณหภูมิ วงจรฐานเวลาจริง หรือ ตัวต้านทานปรับค่าได้ (digi-pot) เป็นต้น



รูปที่ 2.5 รูปแบบการเชื่อมต่ออุปกรณ์ด้วยบัส I^2C

2.3.1.3 พื้นที่เก็บข้อมูล

ข้อมูลที่ได้จากเซนเซอร์อาจถูกออกแบบให้ถูกส่งออกไปผ่านเครือข่ายโดยไม่มีการจัดเก็บหรือ ทำการจัดเก็บลงในหน่วยความจำภายในโนดเซนเซอร์ก่อนค่อยส่งออกไป การเก็บข้อมูลในโนดเซนเซอร์อาจเก็บไว้ในหน่วยประมวลผลหรือหน่วยความจำภายนอกแบบแฟลช (flash memory) ที่นำมาต่อเพิ่ม เช่น อีอีพรอม (EEPROM) หรือ ไมโครเอสดีการ์ด (micro SD card) แต่การเลือกวิธีเก็บข้อมูลไว้ในโนดเซนเซอร์ต้องพิจารณาพลังงานที่ใช้ในการเก็บข้อมูลนั้นด้วย เช่น การเก็บข้อมูลลงในไมโครเอสดีการ์ดจะใช้พลังงานมากกว่าการเก็บข้อมูลลงหน่วยความจำแบบแฟลชประเภทอื่น

2.3.2 ส่วนตรวจวัดและสั่งการ

ส่วนตรวจวัดและสั่งการเป็นจุดเชื่อมต่อระหว่างระบบอิเล็กทรอนิกส์และข้อมูลทางกายภาพ ข้อมูลที่เราสนใจจากสิ่งแวดล้อมภายนอก เช่น อุณหภูมิ ความชื้นสัมพัทธ์ ความเร็วลม หรือความเป็นกรดด่าง ข้อมูลเหล่านี้จะถูกแปลงให้เป็นพลังงานไฟฟ้า เช่น แรงดันไฟฟ้า กระแสไฟฟ้า ด้วยอุปกรณ์ที่เรียกว่าเซนเซอร์ (sensor) ปัจจุบันความก้าวหน้าในการผลิตระบบไฟฟ้าเครื่องกลจุลภาค (micro-electro-mechanical system) หรือเรียกโดยย่อว่า MEMS สร้างขึ้นจากเทคโนโลยีการผลิตวงจรรวม ทำให้ได้เซนเซอร์ที่มีประสิทธิภาพมากขึ้น ขนาดเล็กลง ใช้พลังงานต่ำ และ มีราคาถูกลง ส่งผลให้การนำเซนเซอร์มาใช้ในเครือข่ายเซนเซอร์ไร้สายได้สะดวกมากขึ้น ในตารางที่ 2.1 แสดงตัวอย่างการนำเซนเซอร์ประเภทต่าง ๆ ไปประยุกต์ใช้งาน เซนเซอร์ที่นำมาใช้งานในส่วนตรวจวัดและสั่งการแยกได้ตามประเภทของการเชื่อมต่อได้ 2 ประเภทดังนี้

2.3.2.1 เซนเซอร์ประเภทแอนะล็อก

เซนเซอร์ประเภทแอนะล็อก (analog sensor) การเชื่อมสัญญาณไฟฟ้าที่ได้จากเซนเซอร์จะต้องต่อวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลก่อน เพื่อแปลงสัญญาณทางไฟฟ้าให้เป็นข้อมูลที่ได้จากการตรวจวัดอาจต้องมีการคำนวณจากสมการที่ผู้ผลิตกำหนดมาให้ ความถูกต้องและความละเอียดของข้อมูลที่ได้ขึ้นอยู่กับจำนวนบิตที่ใช้แทนสัญญาณแอนะล็อก เซนเซอร์ประเภทนี้มักจะมีราคาไม่แพง การทำงานไม่ซับซ้อน แต่ข้อมูลการตรวจวัดอาจมีความคลาดเคลื่อนได้ง่ายเนื่องจากสัญญาณแอนะล็อกที่ได้ถูกรบกวนจากสัญญาณภายนอกอื่นได้

2.3.2.1 เซนเซอร์ประเภทดิจิทัล

เซนเซอร์ประเภทดิจิทัล (digital sensor) ภายในมีวงจรการแปลงสัญญาณแอนะล็อกให้เป็นสัญญาณดิจิทัลภายในตัวเซนเซอร์ ลดความผิดพลาดของสัญญาณจากระยะสายสัญญาณและความถี่

ที่มารบกวนจากภายนอก สามารถเชื่อมต่อกันได้โดยตรงโดยใช้ ช่องเชื่อมต่อดิจิทัล ช่องเชื่อมต่ออนุกรม บัสเอสพีไอ หรือ บัส I^2C เป็นต้น การใช้งานจะต้องศึกษารูปแบบชุดคำสั่งตั้งค่า สั่งงาน และอ่านค่าตามผู้ผลิตที่กำหนดไว้ เซนเซอร์ประเภทนี้จะมีความแม่นยำของการตรวจวัดที่ดีกว่าเซนเซอร์ประเภทแอนะล็อก แต่ก็อาจมีราคาที่แพงกว่า มีความซับซ้อนในการใช้งานที่มากกว่า

ตารางที่ 2.1 เซนเซอร์ที่นำไปใช้กับงานด้านต่าง ๆ

เซนเซอร์	ข้อมูลที่ตรวจวัด	การนำมาประยุกต์ใช้งาน
ความเร่ง	ความเร่งในระนาบ 2 มิติ หรือ 3 มิติ	- การเฝ้าระวังและดูแลผู้ป่วยผู้สูงอายุภายในที่พัก - การตรวจวัดการระเบิดของภูเขาไฟ - การตรวจวัดความผิดปกติระบบรางและล้อของรถไฟ - การติดตามสินค้าที่แตกง่ายในการขนส่ง
ความชื้นสัมพัทธ์	ความชื้นสัมพัทธ์ในอากาศ	- การทำเกษตรกรรม เพาะปลูก - ตรวจวัดสิ่งแวดล้อม
อุณหภูมิ	ข้อมูลอุณหภูมิ	- การทำเกษตรกรรม เพาะปลูก เลี้ยงสัตว์ - การตรวจวัดสิ่งแวดล้อม - การตรวจวัดคุณภาพของน้ำ
พีเอช (pH)	ค่าความเป็นกรด-ด่าง	- การเลี้ยงสัตว์น้ำ - การตรวจวัดคุณภาพของน้ำ
อินฟราเรดแบบพาสซีฟ (passive infar-red)	การแผ่รังสีอินฟราเรดจากวัตถุ	- การรักษาความปลอดภัย
แรงกด	แรงกระทำกับเซนเซอร์	- การตรวจวัดในอาคาร ถนน หรือ สะพาน
แม่เหล็ก	วัตถุประเภทโลหะ	- การจราจร - ระบบจอดรถอัจฉริยะ
การสั่น	แรงจากการสั่นไหว การเคลื่อนไหว	- งานรักษาความปลอดภัย - งานตรวจวัดแผ่นดินไหว
รังสีอัลตราไวโอเลต	ความเข้มของรังสีอัลตราไวโอเลต	- การตรวจวัดสิ่งแวดล้อม - การทำเกษตรกรรม เพาะปลูก
ความเร็วลม (anemometer)	ความเร็วลมที่พัด	- การตรวจวัดสิ่งแวดล้อม

จุดเด่นและข้อสังเกตของเซนเซอร์แต่ละประเภทดังแสดงในตารางที่ 2.2 การเลือกใช้ให้เหมาะสมกับงานหรือข้อจำกัดของระบบที่มีจะทำให้การทำงานของโนดเซนเซอร์ทำงานได้มีประสิทธิภาพและมีระยะเวลาที่ทำงานได้ยาวนานมากขึ้นด้วย

นอกจากเซนเซอร์แล้วส่วนตรวจวัดและสั่งการอาจมีการเชื่อมต่อกับอุปกรณ์อื่น เช่น มอเตอร์ (motor) รีเลย์ (relay) หรือ สัญญาณเตือน (alarm) ซึ่งจะใช้เป็นส่วนที่ทำหน้าที่สั่งการเพื่อตอบสนองแจ้งเตือนเหตุการ์ที่เกิดขึ้นตัวอย่าง เช่น โนดเซนเซอร์ที่นำใช้งานในการเพาะเลี้ยงสัตว์น้ำตรวจพบว่าขณะนี้ปริมาณออกซิเจนที่ละลายในน้ำลดต่ำลงอาจเป็นอัตรายต่อสัตว์ที่เพาะเลี้ยง จึงสั่งงานไปยังชุดควบคุมมอเตอร์เพื่อหมุนกังหันตีน้ำเพิ่มปริมาณออกซิเจนในน้ำ ส่วนสั่งการนี้อาจถูกพิจารณาติดตั้งให้กับบางโนดในเครือข่ายเท่านั้น เนื่องจากการสั่งงานไปยังอุปกรณ์อื่นอาจจำเป็นต้องมีการต่อสายไฟหรือใช้พลังงานจากแหล่งจ่ายไฟหลักในการทำงานของอุปกรณ์ที่นำมาต่อพ่วง

ตารางที่ 2.2 การเปรียบเทียบเซนเซอร์ประเภทแอนะล็อกและเซนเซอร์ประเภทดิจิทัล

เซนเซอร์ประเภทแอนะล็อก	**เซ็นเซอร์ประเภทดิจิทัล**
ข้อมูลที่ได้จากการวัดอยู่ในรูปของความต่างศักย์ไฟฟ้า ปริมาณกระแสไฟฟ้า หรือ รูปแบบสัญญาณพัลส์ (pulse)	ข้อมูลที่ได้จากการวัดอยู่ในรูปแบบข้อมูลตัวเลข ค่าจริง “1” หรือ เท็จ “0”
ต้องการวงจรแปลงสัญญาณสำหรับการเชื่อมต่อ	ไม่ต้องการวงจรพิเศษสำหรับเชื่อมต่อ
ข้อมูลหลังจากแปลงค่าต้องนำมาตีความหมายเพื่อให้ได้ค่าข้อมูลที่ต้องการวัดแท้จริง	ไม่ต้องการการแปลความหมายของข้อมูลที่ได้
การเชื่อมต่อถูกรบกวนโดยสัญญาณภายนอกได้ง่าย อาจต้องปรับตั้งค่าเพื่อให้ทำการวัดได้อย่างถูกต้อง	การเชื่อมต่อมีโพรโทคอลสำหรับการสื่อสาร หรือรูปแบบคำสั่งที่กำหนดโดยผู้ผลิต ผ่านบัสเอสพีไอ บัส I^2C หรือ พอร์ตอนุกรม
ราคาต่ำกว่าเซนเซอร์ประเภทดิจิทัล	ราคาสูงกว่าเซ็นเซอร์ประเภทแอนะล็อก

2.3.3 ส่วนสื่อสารไร้สาย

การสื่อสารไร้สายเป็นส่วนที่สำคัญของการใช้งานเครือข่ายเซนเซอร์ไร้สาย การสื่อสารของโนดเซนเซอร์ภายในเครือข่ายใช้ในการแลกเปลี่ยนข้อมูลระหว่างโนดหรือการสื่อสารกับโนดฐานที่เชื่อมต่ออินเทอร์เน็ตเกตเวย์ เทคโนโลยีสื่อสารไร้สายที่นำมาใช้ ได้แก่ คลื่นวิทยุ แสงอินฟราเรด (infra-red) คลื่นเสียงความถี่สูง (ultra-sound) และ การเหนี่ยวนำสนามแม่เหล็ก (inductive fields) แต่การสื่อสารโดยใช้คลื่นวิทยุยังคงเป็นแนวทางที่เหมาะสมสำหรับเครือข่ายเซนเซอร์ไร้สาย [16] เนื่องจากไม่มีข้อจำกัดเรื่องแนวระนาบของการรับส่งสัญญาณวิทยุ เครื่องรับส่งสัญญาณวิทยุในปัจจุบันมีการพัฒนาให้ใช้พลังงานต่ำ อัตราการรับส่งข้อมูลและระยะทางในการสื่อสารได้รับการพัฒนาให้ดีขึ้นกว่าเดิม การเลือกย่านความถี่ที่นำมาใช้ก็เป็นส่วนสำคัญเนื่องจากคลื่นความถี่จะมีหน่วยงานของแต่ละประเทศที่ทำหน้าที่กำกับดูแล การใช้งานย่านความถี่ของเครือข่ายเซนเซอร์ไร้สายในอดีตมีการใช้งานย่านความถี่ที่แตกต่างกันไปตามประเทศที่มีการใช้งาน แต่ในปัจจุบันมีการเปลี่ยนมาใช้ย่านความถี่สาธารณะสำหรับงานด้านอุตสาหกรรม วิทยาศาสตร์ และการแพทย์ (Industrial scientific and medical) หรือเรียกว่าย่านความถี่ ISM Band มากขึ้น สามารถใช้งานได้โดยไม่ต้องขออนุญาตก่อน แต่อุปกรณ์นำมาที่ใช้งานต้องปฏิบัติตามข้อตกลงการใช้งานย่านความถี่ อย่างไรก็ตามการเลือกใช้คลื่นความถี่ใดมาใช้งานอาจต้องศึกษาข้อกำหนดและกฎหมายของแต่ละประเทศเพิ่มเติมเพื่อให้สอดคล้องกับการกำกับดูแลของหน่วยงานในประเทศนั้นด้วย

เครื่องรับส่งสัญญาณวิทยุที่ใช้ในส่วนสื่อสารไร้สายจะเชื่อมต่อกับส่วนประมวลผลผ่านพอร์ตอนุกรมหรือบัสเอสพีไอของไมโครคอนโทรลเลอร์ การสื่อสารไร้สายเป็นส่วนที่ใช้พลังงานมากส่วนหนึ่ง จึงมีงานวิจัยที่นำเสนอการจัดการรูปแบบและวิธีการสื่อสารที่มีประสิทธิภาพมากยิ่งขึ้นสำหรับงานเครือข่ายเซนเซอร์ไร้สาย เพื่อให้การสื่อสารของโนดเซนเซอร์ใช้พลังงานต่ำที่สุด ตลอดจนยืดระยะการทำงานของโนดเซนซอร์อีกด้วย ตัวอย่างมาตรฐานการสื่อสารไร้สายสำหรับเครือข่ายที่ใช้พลังงานต่ำที่มีการนำเสนอ ได้แก่ มาตรฐานซิกบี มาตรฐาน IEEE 802.15.4 และ มาตรฐาน WirelessHART

2.3.4 ส่วนควบคุมพลังงานและแบตเตอรี่

ส่วนควบคุมพลังงานและแบตเตอรี่เป็นจุดเชื่อมต่อแหล่งพลังงานกับโนดเซนเซอร์ ประกอบด้วยแบตเตอรี่ วงจรรายงานการใช้พลังงาน วงจรเพิ่มแรงดันไฟฟ้า วงจรรักษาแรงงดันไฟฟ้า ทำหน้าที่จ่ายพลังงานไฟฟ้าให้โนดเซนเซอร์และอุปกรณ์ที่ต่อพ่วง รายงานการใช้พลังงานและปริมาณพลังงานที่เหลือในแบตเตอรี่ ตลอดจนการควบคุมการประจุไฟซ้ำให้กับแบตเตอรี่ด้วย ข้อมูลพลังงานจะถูกส่งให้ส่วนประมวลผลเพื่อตัดสินใจรูปแบบการใช้งานอุปกรณ์ที่ติดตั้งบนโนดเซนเซอร์ในขณะนั้น เช่น การปิดเปิดอุปกรณ์เชื่อมต่อ การเปลี่ยนระยะเวลาการตรวจวัดหรือการส่งข้อมูล หรือ ระยะเวลา

การทำงานในโหมดหลับ เพื่อทำให้โนดเซนเซอร์ทำงานได้อย่างมีประสิทธิภาพ และ มีระยะเวลาการทำงานมากที่สุดต่อการเปลี่ยนแบตเตอรี่ในแต่ละรอบ การเลือกใช้แบตเตอรี่มาใช้งานสามารถพิจารณาได้จากคุณลักษณะของแบตเตอรี่ตามตารางที่ 2.3

ตารางที่ 2.3 คุณลักษณะของแบตเตอรี่

คุณสมบัติ	รายละเอียด
ขนาดความจุ	จำนวนพลังงานไฟฟ้าที่แบตเตอรี่เก็บไว้หลังจากการประจุไฟจนเต็ม แบตเตอรี่แต่ชนิดสามารถในการเก็บพลังงานที่แตกต่างกันไปตามธาตุที่นำมาผลิต หน่วยที่ใช้วัดความจุของแบตเตอรี่มีหน่วยเป็นมิลลิแอมป์ชั่วโมง (mAh)
ความต่างศักย์ปกติ	ความต่างศักย์เฉลี่ยของแบตเตอรี่ขณะใช้งานโดยมีค่าขึ้นกับปฏิกิริยาเคมีภายในของแบตเตอรี่ เช่น แบตเตอรี่ตะกั่วมีความต่างศักย์ 12 โวลต์ หรือ แบตเตอรี่ลิเธียมมีความต่างศักย์ 3 โวลต์
ประเภทแบตเตอรี่	แบตเตอรี่สามารถแบ่งได้เป็น 2 ประเภทดังนี้ 1. แบตเตอรี่ชนิดประจุไฟใหม่ได้ สามารถประจุไฟใหม่ได้หลังจากใช้จนหมด เนื่องจากสารเคมีที่ใช้ทำแบตเตอรี่สามารถย้อยปฏิกิริยาได้โดยใช้เครื่องประจุไฟ เช่น แบตเตอรี่นิเกิล-แคดเมียม (Ni-Cd) แบตเตอรี่นิเกิล-เมลทัลไฮไดรด์ (Ni-MH) หรือ แบตเตอรี่ลิเธียม-ไอออน (Li-Ion) 2. แบตเตอรี่ชนิดประจุไฟใหม่ไม่ได้ สามารถใช้ได้ครั้งเดียว ปฏิกิริยาเคมีที่อยู่ภายในไม่สามารถย้อนกลับได้ เหมาะสำหรับอุปกรณ์ขนาดเล็ก ใช้กำลังไฟฟ้าน้อย เช่น แบตเตอรี่แอลคาไลน์ (alkaline) แบตเตอรี่สังกะสี (zinc)
อัตราการคายประจุ	การคายประจุที่สะสมภายในแบตเตอรี่ที่ไม่มีการใช้งานเป็นปฏิกิริยาที่เกิดขึ้นได้กับแบตเตอรี่ทุกประเภท อัตราการคายประจุขึ้นอยู่กับชนิดของปฏิกิริยาเคมีภายในแบตเตอรี่นั้น อุณหภูมิที่ใช้เก็บรักษา และ ประเภทของแบตเตอรี่ เช่น แบตเตอรี่ชนิดประจุไฟใหม่ไม่ได้จะมีการคายประจุประมาณ 8 – 20% ของปริมาณความจุต่อปี แบตเตอรี่ชนิดประจุไฟใหม่ได้ในอดีตมีอัตราการคายประจุที่มากกว่าแบตเตอรี่ชนิดประจุไฟใหม่ไม่ได้ แต่ในปัจจุบันมีการพัฒนาให้แบตเตอรี่คายประจุลดลง เช่น แบตเตอรี่นิเกิล-เมลทัลไฮไดรด์ หรือ แบตเตอรี่ลิเธียม-ไอออน

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.4 การพัฒนาโปรแกรมประยุกต์สำหรับโนดเซนเซอร์

การนำเครือข่ายเซนเซอร์ไร้สายไปประยุกต์ใช้งาน ผู้ใช้งานจำเป็นต้องพัฒนาโปรแกรมประยุกต์ (application) ขึ้นมาเพื่อสั่งงานโนดเซนเซอร์ยกตัวอย่าง เช่น งานการตรวจวัดของเซนเซอร์ การรับส่งข้อมูลผ่านเครือข่าย การเข้าสู่โหมดหลับและการตื่นขึ้นมาทำงานตามเวลาที่กำหนด โปรแกรมประยุกต์สำหรับโนดเซนเซอร์สามารถใช้ภาษาโปรแกรมระดับล่างจนถึงภาษาโปรแกรมระดับสูง เช่น ภาษาซี/ซี++ (C/C++) ภาษาจาวา (Java) หรือ ภาษาไพธอน (Python) แม้ว่าภาษาโปรแกรมที่ใช้พัฒนาจะเป็นที่คุ้นเคยกับนักพัฒนาแต่การรูปแบบพัฒนาโปรแกรมประยุกต์สำหรับโนดเซนเซอร์มีความแตกต่างจากการพัฒนาโปรแกรมประยุกต์สำหรับคอมพิวเตอร์ เนื่องจากการพัฒนาโปรแกรมสำหรับโนดเซนเซอร์มีความเฉพาะเจาะจง ทรัพยากรที่สามารถใช้งานได้มีอย่างจำกัด ผู้พัฒนาจะต้องเรียนรู้การทำงานของอุปกรณ์ทั้งหมดที่นำมาเชื่อมต่อในระดับเรจิสเตอร์ ตลอดจนการใช้เครื่องมือพิเศษในการตรวจสอบเมื่อเกิดความผิดพลาดของการทำงานเกิดขึ้น แนวทางการพัฒนาโปรแกรมสำหรับโนดเซนเซอร์แบ่งตามรูปแบบการพัฒนาได้เป็น 2 แนวทาง ดังนี้

2.4.1 การพัฒนาโดยใช้ระบบปฏิบัติการ

การพัฒนาโดยใช้ระบบปฏิบัติการเป็นแนวทางการพัฒนาเพื่ออำนวยความสะดวกในการใช้งานอุปกรณ์ต่าง ๆ ที่ติดตั้งบนโนดเซนเซอร์ ผู้ใช้งานไม่ต้องเข้าไปจัดการทำงานในระดับล่างของอุปกรณ์ การพัฒนาอาจใช้ภาษาโปรแกรมภาษาซี/ซี++ แต่อาจมีการพัฒนาภาษาโปรแกรมภาษาใหม่ขึ้นมาใช้งานเพื่อให้สามารถเรียกใช้งานได้ตามลักษณะการทำงานของระบบปฏิบัติการ การทำงานของระบบปฏิบัติการอาจทำงานตามลำดับที่กำหนดไว้หรือทำงานตอบสนองต่อเหตุการที่เกิดขึ้น แม้ว่าการใช้งานระบบปฏิบัติการจะมีความสะดวกมากขึ้น แต่อุปกรณ์ที่ระบบปฏิบัติการรองรับอาจไม่ตรงกับความต้องการ การปรับแต่งหรือใช้งานอุปกรณ์ที่สร้างขึ้นเองจะต้องมีการปรับโปรแกรมประยุกต์ซึ่งผู้ใช้งานจะต้องเข้าใจกลไกการทำงานของระบบปฏิบัติการและการแก้ไขในไฟล์ที่ใช้กำหนดค่าภายในต่าง ๆ ตัวอย่างระบบปฏิบัติการที่ได้รับความนิยมนำมาใช้พัฒนาได้แก่

ระบบปฏิบัติการ TinyOS [17] เป็นระบบปฏิบัติการแรกที่ออกแบบและพัฒนาขึ้นมาเฉพาะสำหรับงานเครือข่ายเซนเซอร์ไร้สาย และอนุญาตให้ผู้พัฒนารายอื่นนำไปพัฒนาต่อได้ โครงสร้างของระบบปฏิบัติการถูกออกแบบให้แบ่งเป็นส่วนโปรแกรม (component-based) ตามหน้าที่การทำงานของระบบ การแบ่งเป็นส่วนโปรแกรมเพื่อให้พัฒนาโปรแกรมประยุกต์ได้อย่างรวดเร็ว มีขนาดเล็ก ในส่วนคลังโปรแกรมมีการจัดเตรียมโพรโทคอลการสื่อสาร โปรแกรมขับอุปกรณ์เซนเซอร์และเครื่องมือการเก็บข้อมูลให้เรียกใช้งาน การทำงานของระบบปฏิบัติการ TinyOS มีรูปแบบการทำงานแบบตอบสนองต่อเหตุการณ์ ซึ่งต่างจาก

การทำงานของระบบปฏิบัติการอื่นที่ใช้การทำงานตามลำดับงานที่จัดไว้ เช่น เมื่อเกิดเหตุการณ์การรับข้อมูลหรือการอ่านค่าของเซนเซอร์เกิดขึ้น ระบบปฏิบัติการจะทำงานตอบสนองต่อเหตุการณ์ที่เกิดขึ้น โดยจะเข้าไปทำงานในส่วนของโปรแกรมที่มีการเขียนรองรับเหตุการณ์นั้นไว้ ระบบปฏิบัติการ TinyOS พัฒนาขึ้นด้วยภาษาโปรแกรมภาษาเนสซี (nesC) ซึ่งมีพื้นฐานมาจากภาษาโปรแกรมภาษาซี การพัฒนาระบบปฏิบัติการเป็นแบบเปิด อนุญาตให้นำไปแก้ไขและพัฒนาต่อยอดได้จึงได้รับความนิยมจากกลุ่มวิจัยมากกว่า 500 กลุ่ม และกลุ่มบริษัทเอกชนที่นำไปพัฒนาใช้กับผลิตภัณฑ์ของตัวเอง

ระบบปฏิบัติการ Contiki's [18] นำเสนอครั้งแรกในปี ค.ศ. 2005 เป็นระบบปฏิบัติการที่ออกแบบขึ้นมาสำหรับงานเครือข่ายเซนเซอร์ไร้สาย การพัฒนาแบบเปิดเช่นเดียวกับระบบปฏิบัติการ TinyOS การทำงานของระบบปฏิบัติมีรูปแบบการทำงานที่ตอบสนองต่อเหตุการณ์ และยังรองรับการทำงานแบบหลายงานในเวลาเดียวกัน (multithread) ตลอดจนการใช้งานหมายเลขไอพีรุ่น 4 (IPv4) และ หมายเลขไอพีรุ่น 6 (IPv6) แนวคิดและกลไกลการทำงานของระบบปฏิบัติการ Contiki's ถูกนำไปประยุกต์ใช้ในงานด้านอุตสาหกรรมอย่างแพร่หลาย นอกจากนี้แนวคิดการใช้หมายเลขไอพีสำหรับอุปกรณ์เครือข่ายเซนเซอร์ไร้สายที่ใช้พลังงานต่ำนำไปสู่การสร้างมาตรฐาน IEFT และ IPSO อีกด้วย

ระบบปฏิบัติลินุกซ์สำหรับระบบสมองกลฝังตัว (Embedded Linux) ถูกนำมาใช้ในอุปกรณ์ระบบสมองกลฝังตัวที่ทำงานด้วยไมโครคอนโทรลเลอร์รุ่น 32 บิต โดยคงแกนของระบบปฏิบัติการลินุกซ์แต่ตัดลดคุณสมบัติหรือเพิ่มคุณสมบัติบางประการเพื่อเหมาะสมกับทำงานบนอุปกรณ์ขนาดเล็กได้ การใช้งานระบบปฏิบัติการที่พัฒนามาจากระบบปฏิบัติการที่ใช้งานบนคอมพิวเตอร์จะมีจุดเด่นเรื่องการพัฒนาซอฟต์แวร์ที่มีลักษณะใกล้เคียงกับการพัฒนาซอฟต์แวร์บนคอมพิวเตอร์ที่นักพัฒนาส่วนใหญ่คุ้นเคย ไม่ต้องเรียนรู้การใช้งานภาษาโปรแกรมใหม่ ตลอดจนการจัดการโพรโทคอลการเชื่อมต่อเครือข่ายแลน (LAN) และอินเทอร์เน็ต เราอาจพบโนดเซนเซอร์ที่ใช้งานระบบปฏิบัติการดังกล่าวทำหน้าที่แทนคอมพิวเตอร์ภายในเครือข่ายเซนเซอร์ไร้สายที่ทำงานเป็นสถานีฐานหรือเกตเวย์ในการเชื่อมต่อกับเครือข่ายอื่นอีกด้วย

ระบบปฏิบัติการอื่น ๆ เช่น ระบบปฏิบัติการ SOS Embedded Operating System หรือเรียกโดยย่อว่า SOS ทำงานบนพื้นฐานการตอบสนองต่อเหตุการณ์เช่นเดียวกัน มีลักษณะเด่น คือ ระบบการทำงานที่สมบูรณ์จะถูกสร้างจากมอดูลขนาดเล็ก โดยมอดูลเหล่านี้สามารถเพิ่มเข้าไปในระบบได้ขณะระบบยังทำงานอยู่ ตลอดจนการรองรับการจัดการหน่วยความจำแบบไดนามิก (dynamic memory management) แต่ปัจจุบัน

ระบบปฏิบัติการ SOS ได้หยุดการพัฒนาไปแล้ว ระบบปฏิบัติการ LiteOS เป็นระบบปฏิบัติการสำหรับงานเครือข่ายเซนเซอร์ไร้สายแบบเปิด สามารถตอบโต้กับผู้ใช้งาน (interactive) และมีรูปแบบการใช้งานคล้ายกับระบบปฏิบัติการยูนิค (Unix) เครื่องมือที่ใช้พัฒนาที่มาพร้อมกับระบบปฏิบัติการ LiteOS ทำให้ผู้พัฒนาสามารถใช้งานโนดเซนเซอร์ได้เหมือนการใช้งานคอมพิวเตอร์ที่ติดตั้งระบบปฏิบัติการดังกล่าวแล้ว [3]

2.4.2 การพัฒนาโดยไม่ใช้ระบบปฏิบัติการ

การพัฒนาโปรแกรมประยุกต์สำหรับโนดเซนเซอร์โดยไม่ใช้ระบบปฏิบัติการมีอัตราส่วนประมาณร้อยละ 44 [19] ของการพัฒนาโปรแกรมประยุกต์สำหรับโนดเซนเซอร์ทั้งหมด วิธีการดังกล่าวที่เป็นวิธีที่พัฒนาโปรแกรมประยุกต์สำหรับงานที่มีการนำไมโครคอนโทรลเลอร์มาใช้งาน การพัฒนาต้องมีความรู้ความเข้าใจในการเขียนโปรแกรมภาษาซีหรือภาษาซี++ สำหรับสั่งงานไมโครคอนโทรลเลอร์เพื่อการจัดการพลังงาน การเชื่อมต่อกับอุปกรณ์ภายนอก หรือ หน่วยความจำในระดับเรจิสเตอร์ของอุปกรณ์ การพัฒนาอาจเขียนโปรแกรมเพื่อให้ทำงานตามลำดับงาน หรือทำงานแบบตอบสนองต่อเหตุการณ์ (event driven) โดยใช้กระบวนการทำงานแบบวนรอบการทำงานร่วมกับการใช้สัญญาณอินเทอร์รัพท์ (interrupt) จากไมโครคอนโทรลเลอร์ เพื่อเข้าไปทำงานตามเหตุการณ์ที่กำหนดไว้ในโปรแกรม การพัฒนาด้วยทางเลือกนี้ในปัจจุบันสามารถนำแพลตฟอร์ม (platform) การพัฒนาไมโครคอนโทรลเลอร์ที่ชื่อว่า อาดูอิโน (Arduino) มาอ้างอิงใช้พัฒนาโปรแกรมประยุกต์สำหรับโนดเซนเซอร์ได้ เนื่องแฟลตฟอร์มอาดูอิโนได้เตรียมฟังก์ชันในส่วนโปรแกรมขับอุปกรณ์ไว้ให้บางส่วนแล้ว ซึ่งจะช่วยลดเวลาและข้อผิดพลาดที่จะเกิดขึ้นในขั้นตอนการเขียนโปรแกรมได้ ข้อดีของการเลือกพัฒนาโดยไม่เลือกใช้ระบบปฏิบัติการ คือ การพัฒนาสามารถเลือกใช้อุปกรณ์ได้อย่างหลากหลายโดยไม่ต้องคำนึงถึงการสนับสนุนอุปกรณ์นั้นจากผู้พัฒนาระบบปฏิบัติการ โปรแกรมประยุกต์ที่พัฒนาขึ้นสามารถปรับแต่งได้ตามความต้องการมากที่สุด แต่การพัฒนาในแนวทางดังกล่าวนี้ยังมีข้อจำกัดเรื่องการบำรุงรักษาโปรแกรม เนื่องจากมีความเฉพาะเจาะจงสูง การแก้ไขข้อผิดพลาดหรือการปรับปรุงประสิทธิภาพการทำงานในภายหลังอาจพบปัญหาได้ หากผู้พัฒนาเริ่มต้นไม่จัดทำเอกสารหรือคำอธิบายโปรแกรมไว้อย่างเพียงพอ

2.5 ตัวอย่างโนดเซนเซอร์ที่นำการใช้ในเครือข่ายเซนเซอร์ไร้สาย

2.5.1 โนดเซนเซอร์ที่พัฒนาขึ้นมาใช้งานโดยเฉพาะ

โนดเซนเซอร์ที่พัฒนาขึ้นเพื่องานวิจัยเรื่องการตรวจวัดคุณภาพแหล่งน้ำ [20] ในประเทศอินเดีย นำเสนอการประยุกต์ใช้เครือข่ายเซนเซอร์ไร้สายสำหรับตรวจวัดคุณภาพในแหล่งน้ำธรรมชาติ ส่วนประมวลผล ฯ เลือกใช้ไมโครคอนโทรลเลอร์ตระกูล PIC16 รุ่น 8 บิต เชื่อมต่อกับเซนเซอร์ประเภทแอนะล็อกสำหรับตรวจวัดค่าพีเอช (pH) เพียงประเภทเดียว ส่วนการสื่อสารเลือกใช้เครื่องรับส่งสัญญาณวิทยุที่ใช้โพรโทคอลซิกบีในการสื่อสารไร้สาย จุดเด่นของการออกแบบดังกล่าวคือ การเลือกใช้งานไมโครคอรโทรลเลอร์ที่มีราคาถูก (ประมาณ 2 ยูโร) ซึ่งมีวงจรแปลงสัญญาณแอนะล็อกเป็นดิจิทัลความละเอียด 10 บิตภายใน จำนวน 11 ช่องสัญญาณ งานวิจัย [21] ได้มีการออกแบบโดยเพิ่มวงจรขยายสัญญาณ และปรับระดับสัญญาณที่ได้จากเซนเซอร์ประเภทแอนะล็อก ข้อจำกัดที่พบในการออกแบบ คือ การไม่สามารถเชื่อมต่อเซนเซอร์ประเภทดิจิทัลได้เนื่องจากไม่ได้เตรียมช่องเชื่อมต่อประเภทนี้ไว้ เครื่องรับส่งสัญญาณวิทยุถูกติดตั้งไม่ถาวรสามารถปรับเปลี่ยนไปใช้เครื่องรับส่งสัญญาณวิทยุอื่นได้

โนดเซนเซอร์ที่พัฒนาขึ้นในงานวิจัยเพื่อใช้ในการเพาะปลูกในเรือนกระจก [22] ในประเทศสเปน นำเสนอการประยุกต์เครือข่ายเซนเซอร์ไร้สายเพื่อตรวจวัดความชื้นในดินของการทำเกษตรกรรม ส่วนประมวลผล ฯ เลือกใช้ไมโครคอนโทรลเลอร์ตระกูล MSP430F1611 ส่วนสื่อสารไร้สายเลือกใช้เครื่องรับส่งสัญญาณวิทยุ CC2420 ตามมาตรฐาน IEEE 802.15.4 จุดเด่นของการออกแบบดังกล่าวคือการใช้พลังงานของโนดเซนเซอร์เนื่องจากการเลือกใช้ไมโครคอนโทรลเลอร์ที่ใช้พลังงานต่ำ สามารถทำงานต่อเนื่องนาน 10 สัปดาห์ ข้อจำกัดของการออกแบบ คือ โนดเซนเซอร์สามารถเชื่อมต่อกับเซนเซอร์ประเภท SDI-12 เท่านั้น และเครื่องรับส่งสัญญาณวิทยุไร้สายถูกติดตั้งลงบนแผงวงจรหลัก ไม่สามารถปรับเปลี่ยนได้

โนดเซนเซอร์ CoSeN (Configurable Sensor Node) [23] ในงานวิจัยของมหาวิทยาลัยแมรีแลนด์ บัลติมอร์เคาน์ (University of Maryland, Baltimore County) ประเทศสหรัฐอเมริกา นำเสนอการออกแบบโนดเซนเซอร์ที่แยกระบบการทำงานของโนดออกเป็นระบบย่อย ๆ การทำงานในแต่ละส่วนจะมีการควบคุมโดยใช้ไมโครคอนโทรลเลอร์ขนาดเล็ก สามารถสื่อสารเชื่อมต่อกับระบบอื่นได้ผ่านบัส I^2C จุดเด่นของการออกแบบที่แยกระบบออกเป็นส่วนย่อยทำให้สามารถปรับเปลี่ยนระบบที่ต้องการได้ง่าย โดยยังคงระบบการทำงานอื่นไว้ได้ดังเดิม ไม่ต้องออกแบบระบบของโนดเซนเซอร์ขึ้นมาใหม่ทั้งหมด แต่การใช้งานไมโครคอนโทรลเลอร์ควบคุมในระบบย่อย ๆ อาจทำให้เกิดปัญหาความซับซ้อนในการทำงาน และการจัดการพลังงานรวมทั้งระบบได้ งานวิจัยดังกล่าวยังเป็น

การศึกษาเบื้องต้น ยังไม่ได้เสนอผลของการทดลองเชื่อมต่อกับเซนเซอร์ หรืออุปกรณ์อื่น ๆ ที่มีการใช้งาน

งานวิจัยการพัฒนาแพลตฟอร์มเปิดสำหรับเครือข่ายเซนเซอร์ไร้สาย [24] ของมหาวิทยาลัยอัลบอร์ ประเทศเดนมาร์ก นำเสนอการออกแบบโนดเซนเซอร์เพื่อใช้ในห้องปฏิบัติการสำหรับมหาวิทยาลัย เลือกใช้ไมโครคอนโทรลเลอร์ dsPIC30f3013 เป็นหน่วยประมวลผลหลัก เชื่อมต่อกับคอมพิวเตอร์ด้วยพอร์ตอนุกรม การสื่อสารไร้สายเลือกเครื่องรับส่งสัญญาณวิทยุ nRF905 ประเภทไม่มีโพรโทคอลการสื่อสารมาให้ นอกจากนี้ยังติดตั้งเครื่องรับส่งสัญญาณบูลทูทเพื่อให้สามารถติดต่อกับอุปกรณ์เคลื่อนที่ไร้สายที่มีอยู่เดิมได้ การเชื่อมต่อสัญญาณภายนอกถูกออกแบบให้สามารถเชื่อมต่อผ่านช่องต่อที่มีขนาดใหญ่จำนวน 21 ช่อง รองรับการต่อสายไฟจากอุปกรณ์อื่นได้โดยตรงไม่ต้องอาศัยช่องต่อพิเศษ ข้อจำกัดของโนดเซนเซอร์ในงานวิจัยชิ้นนี้ คือ วิธีการดาวน์โหลดโปรแกรมด้วยเครื่องโปรแกรม PICkit 2 และ โปรแกรมสำหรับพัฒนาที่มีการจำกัดการทำงานสำหรับรุ่นที่อนุญาตให้ใช้งานโดยไม่มีค่าใช้จ่าย ในการนำไปใช้งานจริงการโปรแกรมโนดเซนเซอร์จะต้องทำด้วยเครื่องโปรแกรม PICkit2 เท่านั้น ซึ่งทำให้ไม่สะดวกต่อการใช้งานและการใช้งานโปรแกรมเพื่อพัฒนามีค่าใช้จ่ายเพิ่มหากต้องการเขียนโปรแกรมที่มีขนาดใหญ่หรือการใช้งานที่สูงขึ้นนอกเหนือจากความสามารถที่มีให้ในรุ่นที่ไม่มีค่าใช้จ่าย

โนดเซนเซอร์ PSU Mote [25] พัฒนาขึ้นโดยกลุ่มเครือข่ายเซนเซอร์ไร้สาย คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ เลือกใช้ไมโครคอนโทรลเลอร์ LPC2103 ตระกูล ARM7 รุ่น 32 บิต เป็นหน่วยประมวลผลหลักร่วมกับเครื่องรับส่งสัญญาณวิทยุ CC2500 ที่ติดตั้งถาวรบนแผงวงจร ใช้งานบนย่านความถี่ 2.4 กิกะเฮิรตซ์ติดตั้งถาวรบนแผงวงจร ช่องทางเข้า/ออกของไมโครคอนโทรลเลอร์ขนาด 8 บิต และ ช่องต่อสัญญาณแอนะล็อกจำนวน 8 ช่อง การดาวน์โหลดโปรแกรมลงบน PSU Mote สามารถทำได้ผ่านพอร์ตอนุกรมโดยใช้โปรแกรม Flash Magic

2.5.2 โนดเซนเซอร์ที่มีจำหน่ายทั่วไป

โนดเซนเซอร์ Mica 2 [26] Mica Z [27] และ IRIS Mote [28] ใช้ไมโครคอนโทรลเลอร์ตระกูล AVR รุ่น 8 บิต ทำหน้าที่เป็นหน่วยประมวลผลกลางมีจุดเด่นของหน่วยประมวลผลตระกูล AVR คือ การมีหน่วยความจำแบบแฟลชขนาดใหญ่เมื่อเทียบกับหน่วยประมวลผลตระกูลอื่น โนดเซนเซอร์ดังกล่าวมีหน่วยความจำขนาด 128 กิโลไบต์ เครื่องรับส่งสัญญาณวิทยุใช้คลื่นความถี่ย่าน 433/916 เมกะเฮิรตซ์ หรือ 2.4 กิกะเฮิรตซ์ เครื่องส่งและรับสัญญาณวิทยุจะถูกติดตั้งถาวรด้านหลังของแผงวงจรไม่สามารถปรับเปลี่ยนได้ ไม่มีการติดตั้งเซนเซอร์บนแผงวงจรหลัก การตรวจวัดจะต้องติดตั้งเซนเซอร์เพิ่มลงบนแผงวงจรเซนเซอร์ (sensor board) แล้วนำมาติดตั้งบนแผงวงจรหลัก แหล่ง

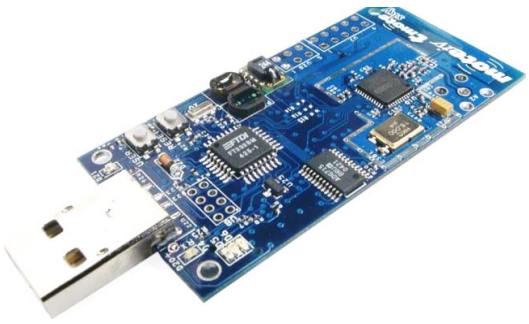
พลังงานสามารถใช้แบตเตอรี่ขนาด AA จำนวน 2 ก้อน ในอดีตโนดเซนเซอร์ดังกล่าวได้รับความนิยมจากผู้วิจัยและรองรับการพัฒนาโปรแกรมประยุกต์โดยใช้ระบบปฏิบัติการสำหรับงานเครือข่ายเซนเซอร์ไร้สาย

รูปที่ 2.6 Mica 2 [26]

รูปที่ 2.7 IRIS Mote [28]

โนดเซนเซอร์ TelosB [29] และ Tmote Sky [30] ใช้ไมโครคอนโทรลเลอร์ตระกูล MSP430 รุ่น 16 บิต ทำหน้าที่เป็นหน่วยประมวลผลมีจุดเด่นเรื่องการใช้พลังงานและการใช้งานเรจิสเตอร์ขนาด 16 บิต ติดตั้งเซนเซอร์วัดอุณหภูมิและความชื้นสัมพัทธ์ลงบนแผงวงจรหลัก สามารถเชื่อมต่อกับคอมพิวเตอร์ด้วยพอร์ตยูเอสบี เครื่องรับส่งสัญญาณวิทยุใช้คลื่นความถี่ย่าน 2.4 กิกะเฮิรตซ์ โดยเครื่องรับส่งสัญญาณวิทยุและเสาอากาศถูกสร้างพร้อมติดตั้งถาวรบนแผงวงจรหลักไม่สามารถปรับเปลี่ยนได้ การเชื่อมต่อกับเซนเซอร์ หรืออุปกรณ์ภายนอกทำได้ไม่สะดวกมากนัก เนื่องจากมีการออกแบบสร้างเชื่อมต่อช่องสัญญาณจากแผงวงจรหลักออกมาให้เพียง 6-10 ช่อง เท่านั้นซึ่งน้อยเกินไปสำหรับการนำมาเชื่อมต่ออุปกรณ์อื่น ๆ สามารถใช้แหล่งพลังงานจากแบตเตอรี่ขนาด AA จำนวน 2 ก้อน และรองรับการพัฒนาโปรแกรมประยุกต์โดยใช้ระบบปฏิบัติการสำหรับงานเครือข่ายเซนเซอร์ไร้สาย

รูปที่ 2.8 TelosB [29]



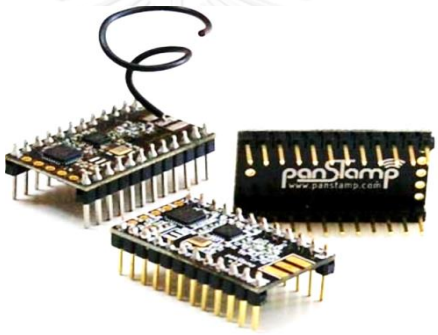
รูปที่ 2.9 Tmote Sky [30]

โนดเซนเซอร์ WESENSE [31] ถูกสร้างขึ้นจากแนวคิดการแบ่งระบบการทำงานในโนดเซนเซอร์ออกเป็นระบบย่อย ๆ ได้แก่ ระบบประมวลผล ระบบสื่อสารไร้สาย ระบบพลังงาน และระบบตรวจวัด โดยพัฒนาขึ้นมาบนแผงวงจรที่แยกจากกัน การใช้งานจะนำแผงจงจรของแต่ละระบบซ้อนกันเป็นชั้น ๆ WISENSE เลือกใช้ไมโครคอนโทรลเลอร์ MSP430G2955 เป็นหน่วยประมวลผลหลัก และ เครื่องรับส่งสัญญาณวิทยุ CC2520 ทำงานบนมาตรฐาน IEEE 802.15.4 จุดเด่นด้านการออกแบบทางฮาร์ดแวร์ของ WISENSE คือการนำแผงวงจรมาต่อซ้อนกันได้ อนุญาตให้ผู้ใช้งานสามารถนำแผงวงจรที่ออกแบบเองมาติดตั้งเพิ่มได้ การพัฒนาซอฟต์แวร์บน WISENSE เลือกพัฒนาโดยไม่ใช้ระบบปฏิบัติการสำหรับเครือข่ายเซนเซอร์ไร้สาย โดยพิจารณารูปแบบการใช้งานโดยวิธีการวนรอบ และตรวจสอบเหตุการณ์ต่าง ๆ ที่เกิดขึ้นโดยการใช้ตัวบ่งชี้ (flag) นอกจากนั้นผู้ผลิตจะเตรียมโปรแกรมขับอุปกรณ์ และส่วนต่อประสานโปรแกรมประยุกต์สำหรับการใช้พัฒนาโปรแกรมประยุกต์ของผู้ใช้ ทั้งนี้การพัฒนาโปรแกรมบน WISENSE ยังมีความซับซ้อนในการศึกษาและใช้งานการเรียกใช้โปรแกรมขับอุปกรณ์และส่วนต่อประสานโปรแกรมประยุกต์ ข้อมูลที่จัดเตรียมไว้ให้ในเว็บไซต์ค่อนข้างซับซ้อนและไม่ครบถ้วน การทำความเข้าใจสำหรับผู้ที่ไม่คุ้นเคยกับการใช้งานไมโครคอนโทรลเลอร์ตระกลู MSP430 ทำได้ค่อนข้างยากเมื่อเทียบกับไมโครคอนโทรลเลอร์ตระกูล AVR นอกจากนี้ประเด็นด้านซอฟต์แวร์แล้วนั้นยังมีข้อสังเกตุสำหรับการใช้งาน เช่น การต้องการวงจรภายนอกเพื่อเชื่อมต่อกับคอมพิวเตอร์ การใช้งานเครื่องโปรแกรมจากภายนอก และการเชื่อมต่อแหล่งพลังงานหรือแบตเตอรี่ เป็นต้น

โนดเซนเซอร์ panStamp [32] การพัฒนาดำเนินโครงการประเภทเปิดเผยรหัสต้นฉบับ (open source) โดยมีจุดประสงค์เพื่อสร้างอุปกรณ์ให้กับนักพัฒนาหรือผู้สนใจใช้งาน เป็นอุปกรณ์แผงวงจรสื่อสารไร้สายขนาดเล็ก เหมาะสำหรับการนำไปใช้งานประยุกต์ที่ใช้พลังงานต่ำ เช่น ระบบอัตโนมัติภายในบ้าน การตรวจวัดสภาพอากาศ และ การควบคุมหุ่นยนต์ เป็นต้น ผู้ใช้ทั่วไปสามารถใช้งานได้โดยการดาวน์โหลดโปรแกรมประยุกต์สำเร็จรูปจากเว็บไซต์ของผู้ผลิตและสั่งซื้อแผงวงจรเซนเซอร์ที่มีจำหน่ายเพิ่มได้ การพัฒนาโปรแกรมประยุกต์สำหรับ panStamp สามารถทำได้โดยสามารถใช้งานซอฟต์แวร์มาตรฐานอาดูอิโนร่วมกับโพรโทคอล SWAP (Simple Wireless Abstract Protocol) และ panStamp Stack ที่ผู้ผลิตนำเสนอได้ ข้อสังเกตการใช้งาน panStamp AVR ด้านฮาร์ดแวร์คือ การทำงานของ panStamp AVR ยังต้องการวงจรภายนอกเพื่อให้สามารถทำงานได้เช่น วงจรจ่ายพลังงาน วงจรการแปลงสัญญาณสำหรับเชื่อมต่อกับคอมพิวเตอร์ (USB-to-serial converter) วงจรดังกล่าวอาจซื้อเพิ่มเติมได้จากผู้ผลิต แต่การใช้งานผู้ใช้จำเป็นต้องเชื่อมต่อสายไฟจาก panStamp AVR ไปยังแผงวงจรที่เพิ่มมาและขั้นตอนการถอดจากแผงวงจรจ่ายพลังงานงานเพื่อไปติดตั้งบนแผงวงจรเชื่อมต่อกับคอมพิวเตอร์เพื่อทำการโปรแกรม อาจเกิดความเสียหายแก่อุปกรณ์ได้หากไม่ระมัดระวังในการใช้งาน นอกจากนั้นเครื่องรับส่งสัญญาณวิทยุที่ติดตั้งบนแผงวงจรหลักไม่

สามารถถอดปรับเปลี่ยนได้ทำให้การนำ panStamp AVR ไปใช้ในบางประเทศไม่ได้เนื่องจากเหตุผลด้านความถี่ของการใช้งาน

รูปที่ 2.10 WISENSE [31]

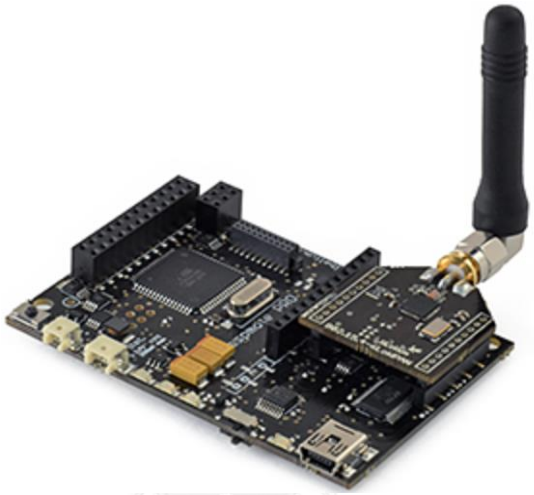


รูปที่ 2.11 panStamp AVR [32]

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

โนดเซนเซอร์ Waspmote [33] ถูกพัฒนาขึ้นในปี ค.ศ. 2009 โดยบริษัท Libelium โครงการพัฒนาเครือข่ายเซนเซอร์แบบเปิดรหัสต้นฉบับในการพัฒนา (เฉพาะซอฟต์แวร์) โดยมีเป้าหมายการใช้งานโนดเซนเซอร์จากอุปกรณ์ที่มีราคาถูก สามารถทำงานได้โดยเองโดยอัตโนมัติและใช้พลังงานจากแบตเตอรี่ แนวคิดเริ่มแรกได้มาจากการใช้งานอาดูอิโนและเครื่องรับส่งสัญญาณวิทยุ Xbee แต่ผู้ผลิตได้พิจารณาถึงข้อจำกัดการใช้งานของบอร์ดอาดูอิโนจึงได้เลือกพัฒนาฮาร์ดแวร์ขึ้นมาใช้งานเองทั้งหมดโดยออกแบบที่มุ่งเน้นฮาร์ดแวร์ที่ใช้พลังงานต่ำและสามารถปรับเปลี่ยนได้ตามความต้องการ ด้านฮาร์ดแวร์ของ Waspmote ถูกออกแบบให้แยกเป็นบอร์ดหลักและบอร์ดเซนเซอร์ โดยบอร์ดหลักประกอบด้วยส่วนประมวลผล ส่วนควบคุมพลังงานและแบตเตอรี่ และ ส่วนสื่อสารไร้สาย รองรับการถอดและติดตั้งเครื่องรับส่งสัญญาณวิทยุไร้สายจากบริษัท Digi ได้หลายประเภท Waspmote ยังรองรับการใช้หน่วยความจำประเภทไมโครเอสดีการ์ดด้วย ในส่วนบอร์ดเซนเซอร์ผู้ผลิตเตรียมบอร์ดเซนเซอร์และเซนเซอร์ไว้ 10 ประเภทสำหรับการใช้งาน ตัวอย่างเช่น บอร์ดเซนเซอร์ตรวจวัด

สิ่งแวดล้อม บอร์ดเซนเซอร์สำหรับงานเกษตร บอร์ดเซนเซอร์สำหรับที่จอดรถ เป็นต้น ด้านซอฟต์แวร์ ผู้ผลิตมีการเตรียมตัวอย่างการใช้งานและโปรแกรมขับอุปกรณ์สำหรับอุปกรณ์ที่ติดตั้งลงในบอร์ดเซนเซอร์แต่ละประเภท การพัฒนาโปรแกรมประยุกต์และเครื่องมือที่ใช้งานมีรูปแบบความคล้ายกับรูปแบบการพัฒนาโปรแกรมประยุกต์สำหรับอาดูอิโน นอกจากนี้ Waspmote ยังมีความสามารถพิเศษในการดาวน์โหลดโปรแกรมประยุกต์ผู้ใช้ผ่านอากาศ (over-the air programming) เพื่ออำนวยความสะดวกในการใช้งานอีกด้วย



รูปที่ 2.12 Waspmote พร้อมเครื่องรับส่งสัญญาณวิทยุไร้สาย [33]

ผู้วิจัยมีโอกาสได้ทดลองใช้งาน Waspmote จริงพบว่าการใช้งานมีจุดเด่นเรื่องการรวมอุปกรณ์ที่จำเป็นในการใช้งานโนดเซนเซอร์ไว้ในแผงวงจรเดียว ตัวบอร์ดมีขนาดเล็กมาก อุปกรณ์อิเล็กทรอนิกส์เกือบทั้งหมดเป็นประเภท SMD (surface mount device) อุปกรณ์อิเล็กทรอนิกส์ถูกติดตั้งทั้ง 2 ด้านของแผงวงจร การพัฒนาโปรแกรมประยุกต์ผู้ใช้มีความคล้ายคลึงกับการพัฒนาโปรแกรมสำหรับบอร์ดอาดูอิโน แต่การใช้งาน Waspmote ยังมีข้อสังเกตดังต่อไปนี้ ผู้ผลิตพัฒนาโปรแกรมขับอุปกรณ์ที่มีการเรียกใช้ฟังก์ชันการทำงานจำนวนมากและไม่ถูกจัดกลุ่มให้ง่ายต่อการใช้งาน ส่งผลให้เกิดความสับสนได้ในระหว่างการใช้งาน ระยะเวลาในขั้นตอนการแปลโปรแกรม (compile) การดาวน์โหลดโปรแกรมลงบน Waspmote จนถึงการเริ่มต้นทำงานใช้เวลาค่อนข้างมาก ซึ่งเป็นข้อจำกัดและความยุ่งยากในการพัฒนาโปรแกรมประยุกต์ อีกทั้งการเชื่อมต่อของบัสเอสพีไอที่ขาดหายไปทำให้ Waspmote ไม่สามารถเชื่อมต่อกับอุปกรณ์ที่ใช้การเชื่อมต่อด้วยบัสเอสพีไอได้เลย ซึ่งเป็นเหตุให้ไม่สามารถใช้เครื่องรับส่งสัญญาณวิทยุจากผู้ผลิตรายอื่นได้ อีกทั้งถ้าเกิดความเสียหายขึ้นกับไมโครคอนโทรลเลอร์หลักผู้ใช้งานจะไม่สามารถทำการเปลี่ยนไมโครคอนโทรลเลอร์ตัวใหม่ได้เอง ต้องทำการส่งกลับไปให้บริษัทผู้ผลิตเป็นผู้เปลี่ยนให้ ซึ่งอาจมีค่าใช้จ่ายจำนวนมากเกิดขึ้นได้

บทที่ 3
การพัฒนาต้นแบบโนดเซนเซอร์

3.1 แพลตฟอร์มสำหรับการพัฒนาฮาร์ดแวร์ของโนดเซนเซอร์

การพัฒนาฮาร์ดแวร์ของโนดเซนเซอร์มีรูปแบบคล้ายกับการพัฒนาอุปกรณ์อิเล็กทรอนิกส์ที่นำระบบสมองกลฝังตัวมาใช้งานโดยมีหน่วยประมวลผลหรือไมโครคอนโทรลเลอร์เป็นศูนย์กลางในการควบคุมและสั่งงาน การพัฒนาฮาร์ดแวร์ของโนดเซนเซอร์เลือกใช้ไมโครคอนโทรลเลอร์มาใช้เป็นหน่วยประมวลผลเนื่องจากมีคุณสมบัติเด่นในด้านประสิทธิภาพการทำงาน การใช้พลังงาน ขนาด และราคาจำหน่าย แนวทางการพัฒนาฮาร์ดแวร์ยังสามารถอ้างอิงกับรูปแบบการพัฒนาโปรแกรมประยุกต์ซึ่งสามารถแบ่งได้เป็น 2 ประเภท ดังนี้

3.1.1 การพัฒนาฮาร์ดแวร์ที่ใช้ระบบปฏิบัติการ

แพลตฟอร์มของฮาร์ดแวร์ที่มีการออกแบบให้รองรับการใช้งานระบบปฏิบัติการในปัจจุบันมีการพัฒนาขึ้นเป็นบอร์ด ขนาดเล็กลง มีจำนวนให้เลือกนำมาใช้งานมากขึ้น ซึ่งเป็นผลมาจากความก้าวหน้าในการสร้างหน่วยประมวลผลที่ประสิทธิภาพสูงและวงจรที่ทำหน้าพิเศษเพิ่มเข้าไปภายในหน่วยประมวลผล ราคาลดลง สามารถการทำงานได้ใกล้เคียงกับหน่วยประมวลผลของคอมพิวเตอร์ทำให้สามารถทำงานที่มีความซับซ้อนได้มากขึ้น จุดเด่นการนำของฮาร์ดแวร์ประเภทไปประยุกต์ใช้งานนี้ ได้แก่ งานที่มีความซับซ้อนต้องการประสิทธิภาพการประมวลผลสูง งานที่เกี่ยวข้องกับสัญญาณภาพ งานที่ต้องการการเชื่อมต่อไปข้ามยังเครือข่ายประเภทอื่น งานที่ต้องการให้สามารถเข้าใช้งานผ่านเว็บไซต์ หรือ งานที่ต้องการเก็บข้อมูลขนาดใหญ่ เป็นต้น

ระบบปฏิบัติการที่นำมาใช้นิยมพัฒนาโดยมีพื้นฐานมาจากระบบปฏิบัติการลินุกซ์ (Linux) ที่ใช้งานอยู่บนคอมพิวเตอร์ส่วนบุคคลทั่วไป เนื่องจากเป็นระบบปฏิบัติการแบบเปิดเผยรหัสต้นฉบับผู้พัฒนาสามารถแก้ไขปรับแต่งได้ตามความต้องการใช้งาน ตลอดจนไม่เสียค่าใช้จ่ายสำหรับการศึกษาวิจัยและใช้งานในเชิงพาณิชย์ ขั้นตอนการพัฒนาโปรแกรมประยุกต์ผู้ใช้มีรูปแบบเหมือนกับการพัฒนาโปรแกรมบนคอมพิวเตอร์ปกติ สามารถเลือกพัฒนาด้วยภาษาโปรแกรมที่มีให้เลือกใช้งานมากกว่าและเรียกใช้ส่วนต่อประสานโปรแกรมประยุกต์ที่ผู้พัฒนาระบบปฏิบัติการจัดหาไว้ให้ได้ การทำงานที่เกี่ยวข้องกับอุปกรณ์ในระดับเรจิสเตอร์และการจัดการการเชื่อมต่ออุปกรณ์จะมีระบบปฏิบัติการช่วยจัดการทำงานให้เกือบทั้งหมด ตัวอย่างแพลตฟอร์มที่ได้รับความนิยมนำมาใช้งานมีรายละเอียดดังต่อไปนี้

3.1.1.1 ราสเบอรี พาย

ราสเบอรี พาย (Raspberry Pi) ลักษณะคล้ายกับคอมพิวเตอร์ส่วนบุคคลขนาดเล็กไม่มีอุปกรณ์ต่อพ่วงอื่น เริ่มแรกแพลตฟอร์มราสเบอรี พาย ถูกออกแบบเพื่อเป็นเครื่องมือสำหรับใช้งานด้านวิทยาการคอมพิวเตอร์ (computer science) ราคาไม่แพง และ สามารถเชื่อมต่อกับอุปกรณ์ภายนอกได้ เช่น มอเตอร์ จอภาพ และ เซนเซอร์ ผู้ออกแบบยังต้องการเปลี่ยนแนวโน้มที่ผู้ใช้งานคอมพิวเตอร์จะต้องเสียเงินซื้อคอมพิวเตอร์ราคาแพง และต้องการทำให้คอมพิวเตอร์สามารถใช้งานได้อย่างกว้างขวางสำหรับทุกคนมากขึ้น [34] บอร์ดราสเบอรี พาย ที่มีวางจำหน่ายแบ่งได้เป็น 2 แบบ [35] ได้แก่ Model A และ Model B โดยรายละเอียดของบอร์ดในแต่ละแบบและรุ่นของการพัฒนาแสดงในตารางที่ 3.1

ตารางที่ 3.1 รายละเอียดของบอร์ดราสเบอรี พาย Model A และ Model B

Model A	**รุ่นที่ 1**	หน่วยประมวลผลตระกูล ARM ของบริษัท Broadcom BCM2835 มีแรมขนาด 256 เมกะไบต์ ช่องเชื่อมต่อสำหรับใช้งานทั่วไป จำนวน 26 ช่อง พอร์ตยูเอสบีจำนวน 1 ช่อง พอร์ตเอชดีเอ็มไอ (HDMI port) จำนวน 1 ช่อง ช่องสัญญาณวิดิทัศน์คอมโพสิต (composite video port) จำนวน 1 ช่อง ช่องหูฟังขนาด 3.5 มิลลิเมตรจำนวน 1 ช่อง และ ช่องต่อไมโครเอสดีการ์ด แต่ไม่มีช่องเชื่อมต่อเครือข่ายแลน
Model B	**รุ่นที่ 1**	พัฒนาขึ้นมาจาก Model A เพิ่มแรมให้มีขนาด 512 เมกะไบต์ ช่องทางเข้า/ออกสัญญาณจำนวน 40 ช่อง พอร์ตยูเอสบีจำนวน 2 ช่อง และ ช่องต่อ RJ-45 สำหรับเครือข่ายแลนจำนวน 1 ช่อง
	รุ่นที่ 2	พัฒนาขึ้นจากรุ่นที่ 1 เปลี่ยนหน่วยประมวลผลใหม่โดยใช้หน่วยประมวลผล Broadcom BCM2836 ตระกูล ARM Cortex-A7 4 แกนสมอง ทำงานที่ความเร็ว 900 เมกะเฮิรตซ์ แรมขนาด 1 กิกะไบต์
	รุ่นที่ 3	พัฒนาขึ้นจากรุ่นที่ 2 เปลี่ยนหน่วยประมวลผลใหม่โดยใช้หน่วยประมวลผลตระกูล ARM Cortex-A8 64 บิต 4 แกนสมอง ทำงานที่ความเร็ว 1.2 กิกะเฮิรตซ์ เพิ่มความสามารถในการเชื่อมต่อเครือข่ายแลนไร้สาย (Wireless LAN) มาตรฐาน IEEE 802.11n และ เครื่องรับส่งสัญญาณวิทยุบูลทูท รุ่น 4.0 ออกวางจำหน่ายในเดือนกุมภาพันธ์ ค.ศ. 2016

รูปที่ 3.1 ราสเบอรี พาย Model B รุ่นที่ 3 [36]

บอร์ดราสเบอรี พาย เป็นแพลตฟอร์มที่ใช้ระบบปฏิบัติการ ผู้ใช้จำเป็นต้องดาวน์โหลดระบบปฏิบัติการที่เก็บอยู่ในไฟล์ที่มีการบีบอัดเรียกว่า ไฟล์อิมเมจ (image file) แล้วติดตั้งลงในหน่วยความจำแบบไมโครเอสดีการ์ดขนาด 4 กิกะไบต์ ขึ้นไปก่อนในการใช้งานครั้งแรก ขั้นตอนดังกล่าวสามารถทำบนคอมพิวเตอร์จากนั้นจึงนำไมโครเอสดีการ์ดไปติดตั้งที่บอร์ดเพื่อเริ่มใช้งาน การทำงานสามารถต่อจอภาพและคีย์บอร์ดเพื่อใช้งานหรือใช้งานผ่านการโปรแกรมประเภท secure shell โดยเชื่อมต่อกับเครือข่ายแลนผ่านพอร์ตมาตรฐาน RJ45 ที่เตรียมไว้บนบอร์ดโดยไม่ต้องต่อจอภาพ ระบบปฏิบัติการที่นำมาใช้กับบอร์ดราสเบอรี พาย แสดงในตารางที่ 3.2

การพัฒนาโปรแกรมประยุกต์สำหรับบอร์ดราสเบอรี พาย ผู้พัฒนาแนะนำให้ใช้ภาษาโปรแกรมภาษาไพธอน (Python Programming Language) ซึ่งเป็นภาษาโปรแกรมระดับสูง การเขียนรหัสต้นฉบับมีความใกล้เคียงกับการเขียนประโยคในภาษาอังกฤษ สามารถอ่านทำความเข้าใจได้ง่าย ผู้ผลิตยังเตรียมโปรแกรมขับอุปกรณ์ คลังโปรแกรม สำหรับเชื่อมต่อกับอุปกรณ์ภายนอกอีกด้วย ลักษณะของโปรแกรมที่พัฒนาด้วยภาษาไพธอน คือ รหัสต้นฉบับที่พัฒนาเสร็จไม่ต้องมีขั้นตอนการแปลโปรแกรมให้เป็นภาษาเครื่อง (machine code) ก่อนเริ่มทำงาน หากโปรแกรมมีการเขียนไวยากรณ์ที่ผิดพลาดจะไม่ถูกตรวจพบจนกว่าโปรแกรมจะทำงานไปถึงจุดที่มีความผิดพลาดนั้น ซึ่งภาษาโปรแกรมไพธอนได้มีการจัดเตรียมการเพื่อจัดการข้อผิดพลาดนี้ไว้ให้แล้ว นอกจากการพัฒนาโปรแกรมด้วยภาษาโปรแกรมไพธอนแล้วราสเบอรี พาย ยังรองรับการใช้ภาษาโปรแกรมอื่น เช่น ภาษาโปรแกรมภาษาซี/ซี++ ภาษาโปรแกรมภาษาจาวา ภาษาโปรแกรมภาษาซี# (C# Programming Language) และ ภาษาโปรแกรมภาษาเบสิก (Basic Programming Language) อีกด้วย

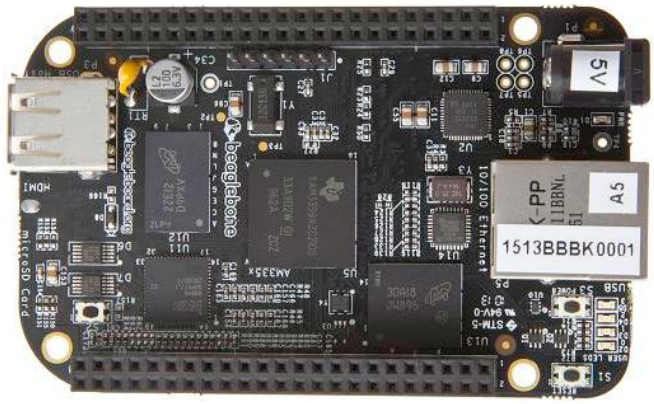
ตารางที่ 3.2 ระบบปฏิบัติการที่นำมาใช้กับบอร์ดราสเบอรี พาย

ระบบปฏิบัติการที่พัฒนาโดยมีพื้นฐานจากระบบปฏิบัติการลินุกซ์	
Raspbian	พัฒนาขึ้นจากระบบปฏิบัติการลินุกซ์ที่มีดิสทริบิวชันที่ชื่อว่า Debian ร่วมกับ ส่วนต่อประสานกราฟฟิกผู้ใช้ (graphic user interface – GUI) ที่ชื่อว่า Lightweight X11 Desktop environment มีชื่อย่อว่า LXDE ระบบปฏิบัติการนี้ได้รับการสนับสนุนอย่างเป็นทางการและต่อเนื่องจากผู้ผลิต (ปัจจุบันมีชื่อรุ่นว่า Rasbian Jessie) สามารถใช้โปรแกรม NOOBS ช่วยในขั้นตอนการดาวน์โหลดและติดตั้งในไมโครเอสดีการ์ด [37]
Ubuntu MATE	พัฒนาขึ้นจากระบบปฏิบัติการลินุกซ์ที่มีดิสทริบิวชันที่มีชื่อว่า Ubuntu ทำให้สามารถใช้เครื่องมือการติดตั้งโปรแกรมได้เหมือนการใช้งานบนบนคอมพิวเตอร์ [38]
ระบบปฏิบัติการอื่น ๆ	
RISC OS Pi	พัฒนามาสำหรับไมโครคอนโทรลเลอร์ตระกูล ARM โดยเฉพาะ มีส่วนต่อประสานกราฟฟิกให้ใช้งานได้ [39]
Windows 10 IoT	พัฒนาขึ้นสำหรับบอร์ดราสเบอรี พาย สามารถติดตั้งโปรแกรมประยุกต์ประเภท Universal App (โปรแกรมประยุกต์ที่สามารถทำงานได้บนคอมพิวเตอร์ โทรศัพท์สมาร์ตโฟน เครื่องเล่นเกม Xbox) [40]

โนดเซนเซอร์ที่สร้างขึ้นจากบอร์ดราสเบอรี พาย จะมีจุดเด่นในด้านการประมวลผล การเชื่อมต่อกับเครือข่ายแลนหรืออินเทอร์เน็ตโดยไม่ต้องซื้ออุปกรณ์เพิ่ม และ การแสดงภาพกราฟฟิกออกทางจอภาพ เช่น โนดเซนเซอร์ที่ใช้รวบรวมข้อมูลภายในเครือข่าย โนดเซนเซอร์ที่ใช้งานเป็นเว็บเซิร์ฟเวอร์ เกตเวย์ของเครือข่าย เป็นต้น ข้อจำกัดการใช้งานของบอร์ดราสเบอรี พาย ได้แก่ การใช้พลังงานที่ค่อนข้างสูงเนื่องจากไม่ได้ออกแบบให้ทำงานในโหมดนอนหลับ การเชื่อมต่อกับอุปกรณ์ภายนอกที่ระบบปฏิบัติการยังไม่รองรับการใช้งานและการใช้งานอุปกรณ์ที่มีระดับสัญญาณสำหรับสื่อสารขนาดเกินกว่า 3.3 โวลต์ รวมถึงค่าใช้จ่ายในการนำมาใช้งานด้วย

3.1.1.2 บีเกิลโบน

บีเกิลโบน (BeagleBone) เป็นแพลฟอร์มการพัฒนาคอมพิวเตอร์ที่สามารถต่อขยายความสามารถในการทำงานได้ ราคาถูก ทำงานโดยใช้ระบบปฏิบัติการลินุกซ์ การพัฒนาเป็นแบบเปิดเผยรหัสต้นฉบับทั้งฮาร์ดแวร์และซอฟต์แวร์ นำเสนอในเดือนพฤศจิกายน ปี ค.ศ. 2011 โดยกลุ่มนักพัฒนาจากบริษัท Texas Instruments มีชื่อว่า BeagleBoard.org เป้าหมายของโครงการบีเกิลโบน คือ การพัฒนาคอมพิวเตอร์ที่มีประสิทธิภาพสูง สามารถต่อขยายได้ นำให้กับผู้พัฒนานวัตกรรมรุ่นใหม่ ตลอดจนการใช้งานที่งาน โดยมีพื้นฐานจากการใช้งานเว็บเบราว์เซอร์ (browser-based) ร่วมกับภาษาสคริปต์ที่มีชื่อว่า Bonescript และจัดเตรียมโปรแกรมขับอุปกรณ์ที่พัฒนาด้วยโปรแกรมภาษาซีสำหรับควบคุมสั่งงานหน่วยประมวลผลและอุปกรณ์ที่ติดตั้งบนบอร์ด [41] บีเกิลโบนมีการพัฒนาบอร์ดขึ้นอย่างต่อเนื่องตั้งแต่เริ่มโครงการโดยเลือกใช้หน่วยประมวลผลตระกูล ARM Cortex-A8 ของบริษัท Texas Instruments Sitara AM335x มีลำดับการพัฒนาบอร์ดรุ่นต่าง ๆ โดยข้อมูลการพัฒนาปรับปรุงบอร์ดรุ่นต่าง ๆ [42] สำคัญแสดงในตารางที่ 3.3 สำหรับแพลตฟอร์มบีเกิลโบนที่ผู้วิจัยสนใจจะกล่าวโดยเน้นไปที่บอร์ด BeagleBone Black แสดงในรูปที่ 3.2 เนื่องจากบอร์ดดังกล่าวได้รับความนิยมในการนำมาใช้งานจากผู้พัฒนาจากหลายประเทศและได้รับการพัฒนาขึ้นมาเป็นรุ่นล่าสุดในปัจจุบัน



รูปที่ 3.2 บอร์ด BeagleBone Black [43]

ตารางที่ 3.3 ลำดับการพัฒนาฮาร์ดแวร์ของโครงการบีเกิลโบน

บอร์ด	การพัฒนา
BeagleBoard (ค.ศ. 2008)	พัฒนาขึ้นครั้งแรกโดยใช้หน่วยประมวลผลตระกูล ARM Cortex-A8 ความเร็ว 720 เมกะเฮิรตซ์ รองรับสัญญาณภาพความคมชัดสูง (HD video) แต่ยังไม่มีรองรับการเชื่อมต่อกับเครือข่ายคอมพิวเตอร์
BeagleBoard xM (ค.ศ. 2010)	รูปแบบเหมือน BeagleBoard แต่มีการเปลี่ยนหน่วยประมวลผลเป็น รุ่น AM37x ความเร็ว 1 กิกะเฮิรตซ์ แรมขนาด 512 เมกะไบต์ พอร์ตยูเอสบี 4 ช่อง รองรับการเชื่อมต่อเครือข่ายคอมพิวเตอร์
BeagleBone (ค.ศ. 2011)	ขนาดเล็กกว่า BeagleBoard หน่วยประมวลผลความเร็ว 720 กิกะเฮิรตซ์ แรมขนาด 256 เมกะไบต์ รองรับการเชื่อมต่อเครือข่ายคอมพิวเตอร์และการเชื่อมต่อกับอุปกรณ์ในระดับล่าง (เช่น ช่องต่อดิจิทัล 65 ช่อง วงจรแปลงสัญญาณแอนะล็อกเป็นดิจิทัล บัสเอสพีไอ บัส I^2C) แต่ไม่รองรับสัญญาณภาพ
BeagleBone Black (ค.ศ. 2013)	พัฒนาต่อจาก BeagleBone ใช้หน่วยประมวลผลความเร็ว 1 กิกะเฮิรตซ์ แรมประเภท DDR3 ขนาด 512 เมกะไบต์ รองรับการเชื่อมต่อเครือข่ายคอมพิวเตอร์ การเชื่อมต่อกับอุปกรณ์ในระดับล่าง รองรับสัญญาณภาพความคมชัดสูง และ หน่วยความจำแบบ eMMC ขนาด 2 กิกะไบต์

บีเกิลโบนเป็นแพลตฟอร์มที่เลือกใช้ระบบปฏิบัติการลินุกซ์โดยใช้หน่วยความจำแบบไมโครเอสดีการ์ดทำหน้าที่เปรียบเสมือนฮาร์ดดิสก์สำหรับเก็บระบบปฏิบัติการและข้อมูลอื่น ๆ แต่ในบอร์ด BeagleBone Black จะมีการเพิ่มหน่วยความจำประเภท eMMC ติดตั้งลงบนบอร์ดสำหรับใช้เก็บระบบปฏิบัติการแทนไมโครเอสดีการ์ด แต่ยังคงช่องต่อไมโครเอสดีการ์ดไว้ดังเดิมเพื่อใช้สำหรับเก็บข้อมูลที่มีขนาดใหญ่อื่น ๆ การใช้งานบอร์ดบีเกิลโบนโดยเฉพาะบอร์ด BeagleBone Black มีจุดเด่นตรงที่สามารถใช้งานหลังจากแกะออกจากกล่องได้เลยเนื่องจากมีการติดตั้งระบบปฏิบัติการไว้ในหน่วยความจำประเภท eMMC เรียบร้อยแล้ว ผู้ใช้งานยังสามารถเลือกระบบปฏิบัติการลินุกซ์ที่มีให้เลือกหลายดิสทริบิวชัน ระบบปฏิบัติการแอนดรอย (Android) รวมถึงระบบปฏิบัติการลินุกซ์บางรุ่นที่มีการพัฒนาเพื่อการเขียนโปรแกรมแบบเวลาจริง (real time programming) ซึ่งเสียค่าใช้จ่ายในการนำไปใช้งาน สำหรับบอร์ด BeagleBone Black กลุ่มผู้พัฒนามีการเลือกใช้งานระบบปฏิบัติการลินุกส์ที่มีดิสทริบิวชันดังต่อไปนี้

Debian มีชุมชนนักพัฒนาที่เน้นการพัฒนาแบบเปิดเผยรหัสต้นฉบับโดยไม่เกี่ยวข้องกับองค์กรทางการค้าใด ๆ และ สนับสนุนการพัฒนาข้ามแพลตฟอร์ม

Angstrom ถูกใช้อย่างแพร่หลายในอุปกรณ์ที่ใช้ระบบสมองกลฝังตัวสามารถย่อยขนาดระบบปฏิบัติการให้มีขนาดเล็กลงเพื่อให้เหมาะสมกับขนาดของหน่วยความจำที่ใช้งานของอุปกรณ์

Ubuntu มีความใกล้เคียงกับ Debian ได้รับความนิยมสำหรับคอมพิวเตอร์ส่วนบุคคลทั่วไป เหมาะสมสำหรับผู้ใช้งานใหม่และใช้งานได้ง่าย

Arch Linux มีขนาดเล็กและยืดหยุนในการใช้งาน ระบบยอมให้ผู้ใช้สามารถกำหนดควบคุมการทำงานได้อย่างสมบูรณ์ แต่เมื่อเทียบกับระบบปฏิบัติการอื่นในด้านการสนับสนุน อาจมีข้อมูลที่น้อยกว่าสำหรับผู้ใช้ใหม่

การพัฒนาโปรแกรมประยุกต์สำหรับแพลตฟอร์มบีเกิล [44] โบนสามารพัฒนาโปรแกรมประยุกต์ได้เหมือนการพัฒนาโปรแกรมประยุกต์บนคอมพิวเตอร์ทั่วไป เนื่องจากสามารถติดตั้งรับปฏิบัติการที่ใช้งานบนคอมพิวเตอร์ได้ กลุ่มนักพัฒนาแพลตฟอร์มบีเกิลโบนนำเสนอทางเลือกการใช้งานเทคโนโลยีใหม่สำหรับการพัฒนาซอฟต์แวร์และฮาร์ดแวร์บนแพลตฟอร์มซึ่งประกอบ 3 ส่วนดังต่อไปนี้

Node.Js แพลตฟอร์มสำหรับพัฒนาโปรแกรมประยุกต์สำหรับเครือข่าย มีวิธีการใช้งานคล้ายกับการใช้งานจาวาสคริปต์ใช้หลักการทำงานแบบสนองต่อเหตุการณ์ร่วมกับรูปแบบการเข้าถึงการเชื่อมต่ออุปกรณ์แบบไม่กีดกัน (non-blocking I/O model) ทำให้โปรแกรมประยุกต์สามารถใช้งานทรัพยากรบนบอร์ดร่วมกันได้

Cloud9 IDE โปรแกรมสำหรับพัฒนาที่รวมเครื่องมือที่จำเป็นต่อการพัฒนาไว้ในที่เดียว รองรับการพัฒนาด้วยภาษาโปรแกรมจำนวนมากรวมถึงการใช้ Node.Js สามารถเขียนรหัสต้นฉบับ จำลองการทำงาน ตรวจสอบข้อผิดพลาดได้ภายในโปรแกรมเดียวผ่านเว็บเบราว์เซอร์ โดยไม่ต้องติดตั้งโปรแกรมเฉพาะใด ๆ เพิ่มเติมและสามารถเรียกใช้ได้จากเว็บเบราว์เซอร์ที่ติดตั้งบนบอร์ด BeagleBone Black ได้อีกด้วย

BoneScript คลังโปรแกรมสำหรับเรียกใช้งานการและเชื่อมต่อของอุปกรณ์สำหรับบอร์ด BeagleBone Black สำหรับการใช้งานร่วมกับ Node.Js มีรูปแบบการใช้งานคล้ายกับคลังโปรแกรมของอาดูอิโน

เช่น การเรียกใช้งานหลอดแอลอีดี การเชื่อมต่อเซนเซอร์ และมอเตอร์

บีเกิลโบนเหมาะสำหรับโนดเซนเซอร์ที่ต้องการการประมวลผลที่ซับซ้อนและสามารถเชื่อมต่อกับอุปกรณ์ภายนอกได้สะดวก เนื่องจากมีการออกแบบช่องเชื่อมต่อและบอร์ดต่อขยายที่มีชื่อเรียกว่า Cape สำหรับเพิ่มความสามารถในการทำงานให้กับบอร์ด เช่น งานอัตโนมัติภายในบ้านหรือโรงงาน ระบบที่มีการเชื่อมต่อกับเซนเซอร์และเครือข่าย แต่ข้อจำกัดการใช้งานของบีเกิลโบนเช่นเดียวกับแพลตฟอร์มราสเบอรี พาย ได้แก่ เรื่องการใช้พลังงานที่ไม่สามารถทำงานในโหมดนอนหลับ ความซับซ้อนในการจัดการอุปกรณ์ที่ยังระบบปฏิบัติการยังไม่รองรับ และ ราคาจำหน่ายที่ค่อนข้างสูง การเลือกนำมาใช้งานเป็นโนดสำหรับตรวจวัดในเครือข่ายยังไม่เหมาะสมมากนัก

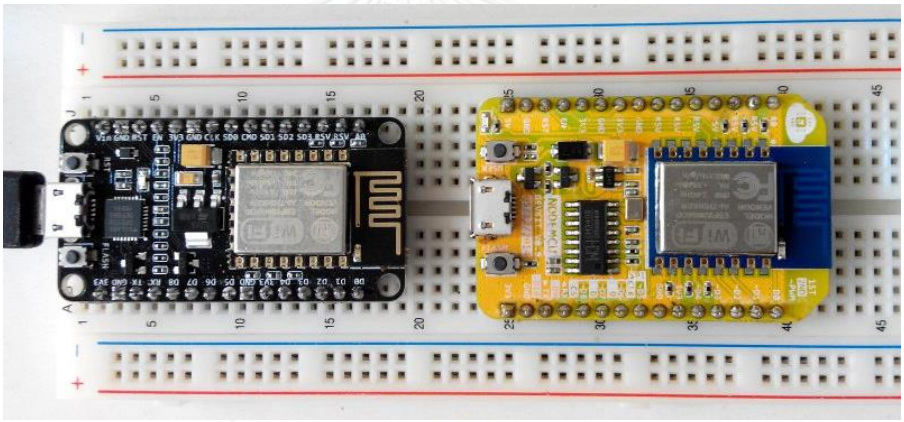
3.1.2 การพัฒนาฮาร์ดแวร์ที่ไม่ใช้ระบบปฏิบัติการ

ฮาร์ดแวร์ที่ไม่ใช้ระบบปฏิบัติการจะมีความซับซ้อนของบอร์ดน้อยกว่าการพัฒนาฮาร์ดแวร์ที่มีการใช้ระบบปฏิบัติการ นิยมนำไมโครคอนโทรลเลอร์ตระกูล AVR, dsPIC หรือ MSP430 มาใช้งานทำงานด้วยสัญญาณนาฬิกาที่ไม่สูงมาก ในอดีตการพัฒนาฮาร์ดแวร์จะเลือกใช้ไมโครคอนโทรลเลอร์รุ่นหนึ่งมาพัฒนา เนื่องจากการพัฒนาโนดเซนเซอร์จะต้องพัฒนาทั้งด้านฮาร์ดแวร์และซอฟต์แวร์ขึ้นมาเองทั้งหมด จุดเด่นของการพัฒนารูปแบบนี้ คือ การทำงานของฮาร์ดแวร์มีความเฉพาะเจาะจงสูง ปรับแต่งได้ตรงตามความต้องการได้มากที่สุด แต่หากต้องการเปลี่ยนหรือย้ายไปใช้ไมโครคอนโทรลเลอร์รุ่นอื่นจำเป็นต้องเปลี่ยนซอฟต์แวร์ในส่วนโปรแกรมประยุกต์และโปรแกรมขับอุปกรณ์ที่พัฒนาขึ้นไว้แล้วทั้งหมด ทำให้การพัฒนาหรือการอัพเกรดอย่างต่อเนื่องทำได้ยาก ปัจจุบันมีการนำเสนอแพลตฟอร์มสำหรับพัฒนาเพื่อลดข้อจำกัดการยึดติดกับอุปกรณ์ตัวใดตัวหนึ่งอย่างที่กล่าวมาข้างต้น ทำให้การพัฒนาฮาร์ดแวร์มีมีทางเลือกในการใช้งานมากขึ้น แพลตฟอร์มการพัฒนาฮาร์ดแวร์ที่มีการไม่ใช้ระบบปฏิบัติการที่น่าสนใจในปัจจุบัน ได้แก่

3.1.2.1 โนดเอ็มซียู

โครงการโนดเอ็มซียู (NodeMCU) [45, 46] เริ่มขึ้นในเดือนตุลาคม ค.ศ. 2014 การพัฒนาเป็นประเภทเปิดเผยรหัสต้นฉบับนำเสนอแพลตฟอร์มสำหรับการพัฒนาทั้งฮาร์ดแวร์และซอฟต์แวร์เพื่อใช้งานร่วมกัน ฮาร์ดแวร์ของโนดเอ็มซียูมีชื่อว่า NodeMCU DevKit พัฒนาจากเครื่องรับส่งสัญญาณวิทยุ ESP8266 ของบริษัท Espressif ที่ใช้เทคโนโลยีแลนไร้สาย (Wireless LAN) ใช้หน่วยประมวลผล Tensillica Xtensa LX106 รุ่น 32 บิต วงจรแปลงสัญญาณการสื่อสารสำหรับการเชื่อมกับพอร์ตยูเอสบีของคอมพิวเตอร์ สามารถเชื่อมต่อกับอุปกรณ์ภายนอกผ่านช่องต่อดิจิทัล บัสเอสพีไอ

บัส I^2C เป็นต้น ด้านซอฟต์แวร์การสามารถพัฒนาได้ 2 แนวทาง ดังนี้ แนวทางแรกการใช้ภาษาสคริปต์ (script language) ที่มีชื่อเรียกว่า Lua การใช้ภาษาสคริปต์สำหรับพัฒนามีจุดเด่นเรื่องความง่ายในการพัฒนา ภาษาสคริปต์มีไวยากรณ์ที่สามารถเข้าใจได้ง่าย แต่การทำงานจะต้องใช้ทรัพยากรเพื่อทำการแปลภาษาที่มากกว่าการพัฒนาด้วยภาษาโปรแกรมอื่น แนวทางที่สองการใช้ภาษาซี/ซี++ ในการพัฒนาร่วมกับโปรแกรม Arduio IDE ซึ่งจะมีส่วนติดตั้งเพิ่ม (plug-in) สำหรับการพัฒนาเพิ่มจากโปรแกรมเดิม ปัจจุบันโนดเอ็มซียูได้พัฒนาขึ้นจากรุ่นที่ 1 เป็นรุ่นที่ 2 [47] ใช้มอดูลเครื่องรับส่งสัญญาณวิทยุ ESP-12E ของบริษัท Al Thinker หน่วยความจำแบบแฟลชขนาด 4 เมกะไบต์ มีการเพิ่มวงจรรักษาแรงดันไฟฟ้าขนาด 3.3 โวลต์ ขาสัญญาณของบัสเอสพีไอเพื่อเชื่อมต่อหน่วยความจำประเภทเอสดีการ์ดและช่องเชื่อมต่อแอนะล็อกสำหรับแปลงสัญญาณเป็นดิจิทัลที่ความละเอียด 10 บิต ปรับปรุงวงจรแปลงสัญญาณการสื่อสารประเภทอนุกรมและเพิ่มปุ่มกดบนแผงวงจรใช้สำหรับการดาวน์โหลดโปรแกรมใหม่



รูปที่ 3.3 NodeMCU รุ่น 2 (ซ้าย) และ NodeMCU รุ่น 1 (ขวา) [47]

จุดเด่นของการใช้งานแฟลตฟอร์มโนดเอ็มซียู [48] ได้แก่ หน่วยประมวลผลขนาด 32 บิต มีขาสัญญาณให้เชื่อมต่อกับอุปกรณ์อื่น ๆ เช่น เซนเซอร์ หลอดแอลอีดี จอแสดงผล สามารถเชื่อมต่อกับคอมพิวเตอร์และใช้พลังงานไฟฟ้าผ่านพอร์ตยูเอสบีได้และเชื่อมต่อกับเครือข่ายแลนไร้สายได้ เมื่อพิจารณาจุดเด่นด้านฮาร์ดแวร์และความสะดวกในการพัฒนาโปรแกรมแพลตฟอร์ม NodeMCU เหมาะสมสำหรับการนำสร้างโนดเซนเซอร์เพื่อทำหน้าที่เป็นเกตเวย์เชื่อมต่อกับเครือข่ายแลนไร้สายหรือเครือข่ายอินเทอร์เน็ตหรือโนดเซนเซอร์ที่สามารถใช้พลังงานได้จากแหล่งพลังงานไฟฟ้าหลักได้อย่างต่อเนื่อง แต่หากพิจารณาการนำมาสร้างเป็นโนดเซนเซอร์ที่ใช้พลังงานจากแบตเตอรี่เพียงเพียงอย่างเดียวอาจมีข้อจำกัดเรื่องพลังงานที่ใช้ของบอร์ดต่อการเปลี่ยนแบตเตอรี่ในแต่ละรอบ โดยผู้พัฒนาอาจต้องนำมาบอร์ดมาใช้ทดสอบการทำงานเองด้วยตัวเองผู้ผลิตไม่ได้แจ้งข้อมูลการใช้พลังงานไฟฟ้าของบอร์ดไว้อย่างชัดเจน

3.1.2.2 อาดูอิโน

อาดูอิโน (Arduino) เป็นการพัฒนาไมโครคอนโทรลเลอร์แบบเปิดเผยรหัสต้นฉบับทั้งการพัฒนาทั้งฮาร์ดแวร์และซอฟต์แวร์เช่นเดียวกัน มีกลุ่มผู้ใช้งานตั้งแต่ระดับเริ่มต้นจนถึงระดับมืออาชีพขนาดใหญ่ทั่วโลก แนวคิดของการพัฒนาอาดูอิโน คือ การทำให้ไมโครคอนโทรลเลอร์สามารถใช้งานได้ง่ายทั้งด้านฮาร์ดแวร์และการพัฒนาซอฟต์แวร์ [49] ด้านฮาร์ดแวร์อาดูอิโนเลือกใช้ไมโครคอนโทรลเลอร์ตระกูล AVR 8 บิต เป็นหน่วยประมวลผล รองรับการทำงานในโหมดนอนหลับ มีช่องต่อสำหรับอุปกรณ์ภายนอก และ รองรับการเชื่อมต่อสัญญาณที่มีระดับสัญญาณ 3.3 โวลต์ และ 5 โวลต์ได้ บอร์ดมาตรฐานมีให้เลือกงานหลายรุ่นโดยแบ่งตามระดับการใช้งานตั้งแต่บอร์ดเริ่มต้น บอร์ดสำหรับสวมใส่ (wearable device) จนถึงระดับ เครื่องพิมพ์ 3 มิติ (3D printer) ภายในบอร์ดมาตรฐานจะประกอบด้วยไมโครคอนโทรลเลอร์หลัก วงจรรักษาระดับแรงดันไฟฟ้า วงจรแปลงสัญญาณอนุกรมสำหรับเชื่อมต่อช่องยูเอสบี ช่องต่ออุปกรณ์ภายนอก รายละเอียดและคุณสมบัติของบอร์ดอาดูอิโนมาตรฐานรุ่นต่าง ๆ [50] แสดงในตารางที่ 3.4

บอร์ด ARDUINO UNO เป็นบอร์ดมาตรฐานที่ได้รับความนิยมนำใช้ศึกษาทดลองการใช้งานไมโครคอนโทรลเลอร์ มีการเตรียมช่องต่อที่สำคัญ เช่น ช่องต่อดิจิทัล ช่องต่อแอนะล็อก พอร์ตอนุกรม บัสเอสพีไอ และ บัส I^2C เป็นต้น ซึ่งครอบคลุมการเชื่อมต่อกับอุปกรณ์ภายนอกที่นำมาต่อพ่วง นอกจากบอร์ดมาตรฐานแล้วแพลตฟอร์มอาดูอิโนยังสามารถนำมาปรับใช้กับไมโครคอนโทรลเลอร์ตระกูล AVR ตัวอื่นได้ ซึ่งนับว่าเป็นจุดเด่นอย่างหนึ่งของแฟลตฟอร์มที่พัฒนาโดยไม่ยึดติดกับไมโครคอนโทรลเลอร์เพียงรุ่นใดรุ่นหนึ่ง

การพัฒนาโปรแกรมประยุกต์ [51] มีรูปแบบการเขียนรหัสต้นฉบับที่แตกต่างจากการเขียนโปรแกรมด้วยภาษาโปรแกรมภาษาซี/ซี++ ทั่วไปเล็กน้อย โดยโปรแกรมทั่วไปจะเริ่มเขียนคำสั่งไว้ในฟังก์ชัน main() ของโปรแกรม แต่การพัฒนาโปรแกรมประยุกต์สำหรับอาดูอิโนมีขั้นตอนการทำงานแสดงในรูปที่ 3.4 โดยมีรูปแบบการเขียนคำสั่งแยกเป็น 2 ส่วน ดังนี้

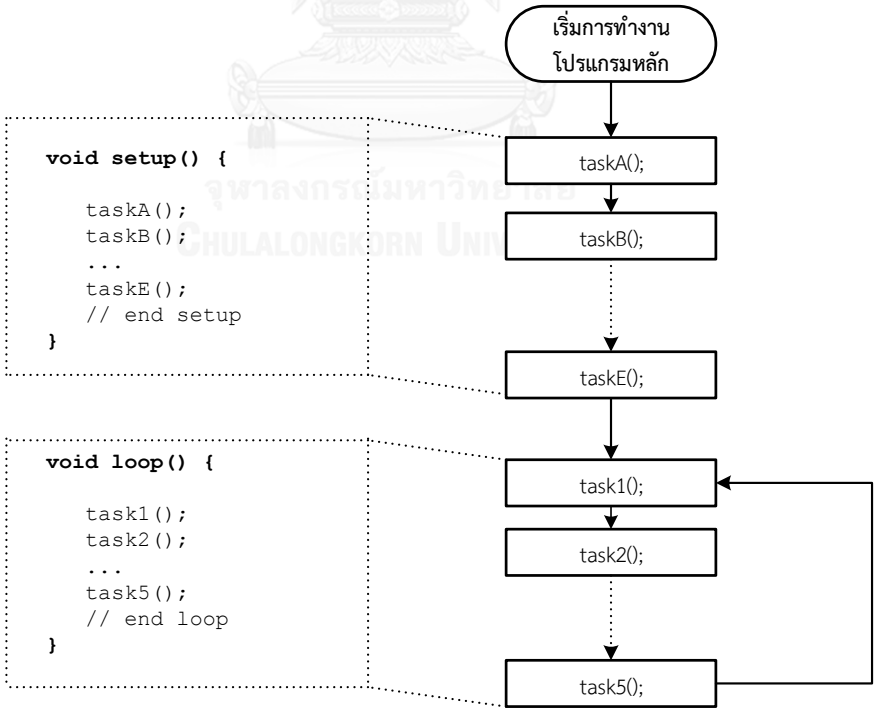
ส่วนที่ 1 ฟังก์ชัน setup() สำหรับเขียนคำสั่งกำหนดค่าการทำงาน เริ่มการเชื่อมต่ออุปกรณ์ คำสั่งเริ่มทำงานของอุปกรณ์ โปรแกรมจะเริ่มทำงานจากคำสั่งถูกเขียนไว้ในส่วนนี้ซึ่งจะทำงานเพียงรอบเดียวโดยไม่มีการวนกลับมาทำอีกตลอดการทำงานของโปรแกรม

ส่วนที่ 2 ฟังก์ชัน loop() สำหรับเขียนชุดคำสั่งการทำงานวนรอบของโปรแกรมโดยจะทำงานตั้งแต่คำสั่งแรกจนถึงคำสั่งสุดท้ายที่เขียนไว้ในส่วนนี้แล้ววนรอบการทำงานกลับมายังคำสั่งแรกอีกครั้งเป็นอย่างนี้เรื่อยไปจนกว่าจะได้รับสัญญาณรีเซต

ตารางที่ 3.4 รายละเอียดของบอร์ดอาดูอิโนแต่ละรุ่น

บอร์ด	ARDUINO UNO	ARDUINO PRO MINI	ARDUINO MEGA 2560	ARDUINO DUE*	ARDUINO GEMMA
ไมโครคอนโทรลเลอร์	ATmega328P (8 bits)	ATmega328 (8 bits)	ATmega2560 (8 bits)	AT91SAM3X8E (32 bits)	ATtiny85 (8 bits)
ประเภท	ระดับเริ่มต้น		ระดับขั้นสูง		อุปกรณ์สวมใส่
หน่วยความจำแฟลช (กิโลไบต์)	32	32	256	512	8
แรม (กิโลไบต์)	2	2	8	96	512 ไบต์
EEPROM (กิโลไบต์)	1	1	4	-	512 ไบต์
สัญญาณนาฬิกา (เมกะเฮิรตซ์)	16	8	16	84	8
แรงดันไฟฟ้าใช้งาน (โวลต์)	5	5	5	3.3	3.3
ช่องสัญญาณดิจิทัล	14	14	54	54	3
ช่องสัญญาณแอนะล็อก	6	6	16	12	1

*ปัจจุบันมีการยกเลิกการผลิตไปแล้วแต่ยังคงจำหน่ายในท้องตลาด



รูปที่ 3.4 แผนผังแสดงลำดับการทำงานของกลุ่มคำสั่งภายในโปรแกรมประยุกต์ผู้ใช้

อาดูอิโนยังพัฒนาโปรแกรมขับอุปกรณ์พร้อมตัวอย่างการเรียกใช้งานสำหรับอุปกรณ์ภายนอก [52] เช่น จอแอลอีดีแบบจุดหรือแบบกราฟฟิก เซนเซอร์ และ การ์ดหน่วยความจำภายนอก โปรแกรมขับอุปกรณ์ที่พัฒนาขึ้นจะถูกสร้างเป็นคลาส (class) ตามรูปแบบการเขียนโปรแกรมเชิงวัตถุ (object oriented programming - OOP) โดยผู้ใช้จะสร้างวัตถุ (object) ของคลาสนั้นขึ้นมาใช้ในโปรแกรม ขั้นตอนการพัฒนารหัสต้นฉบับ การแปลโปรแกรม และ การดาวน์โหลดโปรแกรมลงบนบอร์ด สามารถทำได้โดยใช้โปรแกรม Arduino IDE เพียงโปรแกรมเดียว สามารถใช้งานโดยไม่มีค่าใช้จ่าย หรือ เปลี่ยนไปใช้โปรแกรมอื่นได้ทั้งที่มีค่าใช้จ่ายและไม่มีค่าใช้จ่ายในการใช้งานโปรแกรม เช่น โปรแกรม PROGRAMINO IDE โปรแกรม Visual Studio โปรแกรม Atmel Studio

ขั้นตอนการดาวน์โหลดโปรแกรมลงบนบอร์ดอาดูอิโนไม่ต้องอาศัยเครื่องโปรแกรม (programmer) พิเศษ เนื่องจากแพลตฟอร์มอาดูอิโนมีการพัฒนาเฟิร์มแวร์ที่เป็นโปรแกรมขนาดเล็กมีชื่อเรียกว่า Bootloader ไว้ล่วงหน้า เพื่อใช้ทำหน้าที่สื่อสารกับคอมพิวเตอร์เมื่อต้องการดาวน์โหลดโปรแกรมลงบนบอร์ด โปรแกรม Bootloader จะเริ่มทำงานโดยอัตโนมัติทุกครั้งหลังสัญญาณรีเซต (reset) โดยใช้โพรโทคอล STK500 สื่อสารจากคอมพิวเตอร์เพื่อดาวน์โหลดโปรแกรมผ่านพอร์ตยูเอสบีลงบนบอร์ดผ่านพอร์ตอนุกรมของไมโครคอนโทรลเลอร์

ด้วยจุดเด่นของแพลตฟอร์มอาดูอิโนในด้านฮาร์ดแวร์ที่มีคุณสมบัติสอดคล้องความต้องการการนำมาใช้งานโนดเซนเซอร์ในด้านการเชื่อมต่ออุปกรณ์ ลักษณะการทำงาน ตลอดจนซอฟต์แวร์ที่มีความสะดวกในการพัฒนาสำหรับผู้เริ่มต้นและรูปแบบการพัฒนาโปรแกรมประยุกต์ไม่ได้เปลี่ยนวิธีไปอย่างสิ้นเชิงสำหรับผู้มีประสบการณ์ในการพัฒนาไมโครคอนโทรลเลอร์มาก่อน การเลือกแพลตฟอร์มอาดูอิโนมาพัฒนาเป็นโนดเซนเซอร์สำหรับใช้ตรวจวัดจึงมีความเหมาะสมและน่าสนใจเป็นอย่างยิ่ง

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

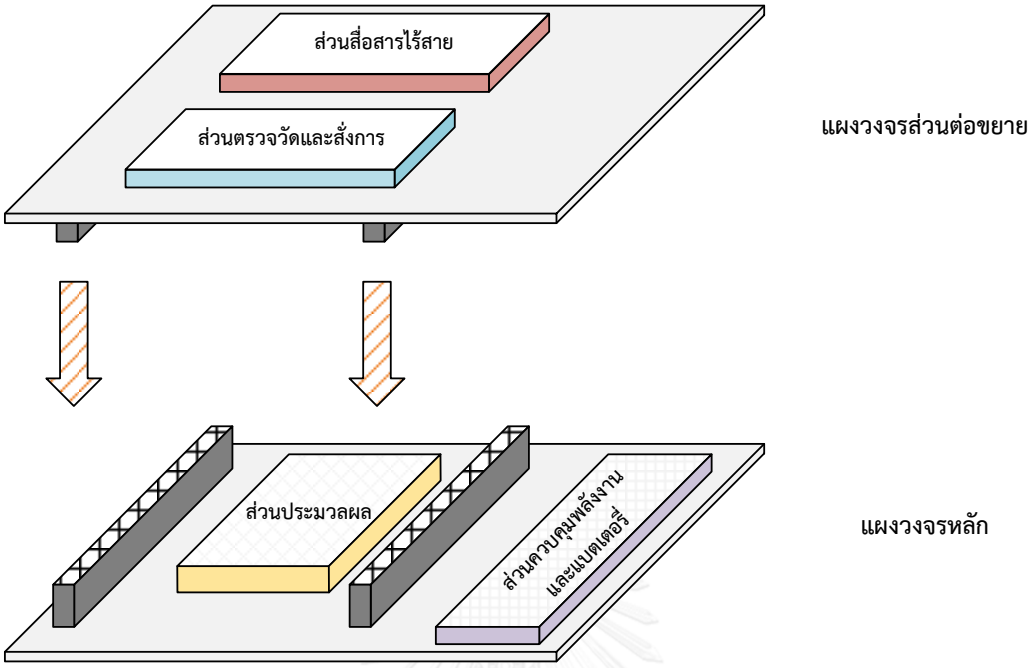
3.2 แนวคิดการพัฒนาต้นแบบโนดเซนเซอร์

การพัฒนาต้นแบบโนดเซนเซอร์ในงานวิจัยนี้เลือกการพัฒนาโดยอ้างอิงกับแพลตฟอร์มอาดูอิโน เนื่องจากเป็นแพลตฟอร์มเปิดรหัสต้นฉบับทั้งฮาร์ดแวร์และซอฟต์แวร์ ไม่มีค่าใช้จ่ายในการนำไปใช้งานทั้งด้านการศึกษาวิจัยและการใช้เชิงพาณิชย์ ลักษณะของโนดเซนเซอร์มีความสอดคล้องกับจุดเด่นของอาดูอิโน คือ ไม่ต้องการการประมวลผลที่ซับซ้อน โนดเซนเซอร์ใช้พลังงานจากแบตเตอรี่ การทำงานจะอยู่ในโหมดนอนหลับเพื่อประหยัดพลังงาน ราคาถูก และสามารถเชื่อมต่อกับอุปกรณ์อื่น ๆ เช่น เซนเซอร์ สวิตช์ มอเตอร์ และ เครื่องรับส่งสัญญาณวิทยุ เป็นต้น การพัฒนาโดยอ้างอิงกับอาดูอิโนทำให้ต้นแบบโนดเซนเซอร์ที่ได้นั้นสามารถใช้งานโปรแกรมขับอุปกรณ์รวมถึงคลังโปรแกรมของบอร์ดอาดูอิโนมาตรฐานได้ การพัฒนาต่อยอดสามารถเปลี่ยนไปใช้บอร์ดอาดูอิโนมาตรฐานรุ่นอื่นหรือบอร์ดที่สร้างขึ้นโดยใช้ไมโครคอนโทรลเลอร์รุ่นอื่นที่รองรับการพัฒนาด้วย

แพลตฟอร์มอาดูอิโนได้ง่ายโดยไม่ต้องเขียนโปรแกรมประยุกต์ของผู้ใช้ใหม่ทั้งหมด ซอฟต์แวร์สำหรับพัฒนาโปรแกรมประยุกต์สำหรับอาดูอิโนยังมีความสะดวกในการใช้งานโดยรวมเครื่องมือที่ใช้ในการพัฒนาพร้อมทั้งโปรแกรมขับอุปกรณ์และตัวอย่างการเรียกใช้งานไว้อย่างครบถ้วน นอกจากนี้ยังมีกลุ่มผู้พัฒนาหลักที่พัฒนาและสนับสนุนแพลตฟอร์มอาดูอิโนอย่างต่อเนื่อง ตลอดจนเว็บไซต์สำหรับแบ่งปันความรู้และวิธีการแก้ปัญหาได้สะดวกอีกด้วย

การไม่เลือกแพลตฟอร์มที่เลือกใช้ระบบปฏิบัติการมาเป็นต้นแบบในงานวิจัยนี้เนื่องจากโนดเซนเซอร์ที่ใช้ในงานวิจัยนี้ต้องการใช้เป็นโนดภายในเครือข่ายไร้สายสำหรับตรวจวัด มีการเชื่อมต่อกับเซนเซอร์ เครื่องรับส่งสัญญาณวิทยุ ด้วยเหตุผลดังกล่าวการเลือกนำบอร์ดราสเบอรี พาย มาใช้งานจะเกิดข้อจำกัดเรื่องช่องเชื่อมต่ออุปกรณ์ภายนอกที่มีจำนวนน้อยและไม่รองรับระดับสัญญาณที่มีขนาด 5 โวลต์ทำให้ต้องเพิ่มวงจรแปลงระดับเข้าไปในการเชื่อมต่อซึ่งอาจทำให้การใช้งานได้ไม่สะดวกมากนัก อีกทั้งปัญหาการไม่สนับสนุนการเชื่อมด้วยบัส OneWire สำหรับเซนเซอร์ขนาดเล็กและราคาถูก การเข้าไปพัฒนาโปรแกรมขับอุปกรณ์เพื่อให้ระบบปฏิบัติการสามารถเรียกใช้งานได้มีความซับซ้อนโดยไม่คุ้มค่ากับการนำอุปกรณ์นั้นมาใช้งาน การใช้พลังงานที่ค่อนข้างมากของบอร์ด ตลอดจนความต้องการของงานที่ไม่ต้องต่อจอภาพเพื่อแสดงผล แม้ว่าบอร์ดบีเกิลโบนจะแก้ปัญหาเรื่องช่องเชื่อมต่ออุปกรณ์ที่มีจำนวนมากกว่าบอร์ดราสเบอรี สามารถเชื่อมต่อเครือข่ายคอมพิวเตอร์ได้สะดวก แต่ด้วยลักษณะงานที่กล่าวไปข้างต้น พลังงานที่บอร์ดต้องการใช้และราคาของบอร์ด BeagleBone Black (ราคาประมาณ 48 เหรียญสหรัฐ) ที่สูงกว่าบอร์ด Arduino UNO (ราคาประมาณ 23 เหรียญสหรัฐ) เกือบสองเท่าทำให้การนำบอร์ดบีเกิลโบนเหมาะสมที่จะนำมาใช้งานที่มีความซับซ้อนมากกว่านี้ การนำมาใช้พัฒนาเป็นโนดเซนเซอร์สำหรับตรวจวัดยังไม่น่าสนใจในขณะนี้ ในส่วนโนดเอ็มซียูแม้ว่าจะมีความน่าสนใจและมีรูปแบบและการใช้งานคล้ายกับอาดูอิโน แต่บอร์ดเอ็มซียูที่จำหน่ายยังขาดวงจรพลังงาน วงจรฐานเวลาจริง การสื่อสารทำได้เพียงการสื่อสารผ่านเครือข่ายแลนไร้สายเพียงอย่างเดียวเท่านั้น ทำให้การใช้งานมีความจำเป็นต้องต่ออุปกรณ์ภายนอกเพิ่มเติม เมื่อดูภาพรวมการใช้งานทั้งหมดแล้วการเลือกแพลตฟอร์มอาดูอิโนมาใช้อ้างอิงในพัฒนาจึงมีความเหมาะสมที่สุด

การพัฒนาต้นแบบโนดเซนเซอร์ในงานวิจัยนี้มีการออกแบบฮาร์ดแวร์และวงจรอิเล็กทรอนิกส์บนบอร์ดตามโครงสร้างของโนดเซนเซอร์ที่ได้กล่าวไว้ในหัวข้อ 2.3 โดยต้นแบบโนดเซนเซอร์จะแยกแผงวงจรออกเป็นสองส่วน คือ แผงวงจรหลักและแผงวงจรส่วนขยาย โดยมีจุดประสงค์เพื่อแยกส่วนที่ใช้ทำงานหลักและส่วนต่อขยายที่สามารถปรับเปลี่ยนได้ตามงานที่นำไปประยุกต์ใช้ การออกแบบให้แยกเป็นสองส่วนทำให้สามารถแยกพัฒนาเฉพาะในส่วนแผงวงจรหลัก หรือแผงวงจรย่อยได้ง่าย โครงสร้างของแผงวงจรหลักและแผงวงจรขยายแสดงในรูปที่ 3.5



รูปที่ 3.5 ภาพโครงสร้างแผงวงจรหลักและแผงวงจรส่วนขยายของต้นแบบโนดเซนเซอร์

3.3 โครงสร้างบอร์ดหลัก

การออกแบบส่วนประกอบภายในบอร์ดหลักของโนดเซนเซอร์จะนำวงจรอิเล็กทรอนิกส์ที่ใช้ในส่วนประมวลผลและส่วนควบคุมพลังงานและแบตเตอรี่ สร้างและติดตั้งรวมไว้บนแผงวงจรเดียวกัน โดยการพัฒนาเลือกพัฒนาแผงวงจรขึ้นมาใช้เองโดยไม่นำบอร์ดอาดูอิโนมาตรฐานที่มีจำหน่ายมาใช้งานโดยตรงเนื่องจากบอร์ดอาดูอิโนมาตรฐานยังมีข้อจำกัดในการทำงานซึ่งขาดวงจรที่จำเป็นต่อการใช้งานเป็นโนดเซนเซอร์ เช่น วงจรฐานเวลาจริงสำหรับการเวลาตั้งปลุกในโหมดนอนหลับ วงจรแปลงระดับแรงดันไฟฟ้าให้สูงขึ้นสำหรับการใช้พลังงานจากแบตเตอรี่ขนาด 1 เซลล์ อีกทั้งขนาดของหน่วยความจำและแรมของบอร์ดมาตรฐานมีขนาดเล็กเมื่อเทียบกับไมโครคอนโทรเลอร์ที่เลือกนำมาใช้ ด้วยข้อจำกัดดังกล่าวการออกแบบแผงวงจรหลักจึงเลือกการพัฒนาขึ้นมาใช้เองโดยมีรายละเอียดของอุปกรณ์ส่วนประกอบต่าง ๆ ดังต่อไปนี้

3.3.1 ไมโครคอนโทรลเลอร์

หน่วยประมวลผลหลักที่เลือกใช้ในโนดต้นแบบ คือ ไมโครคอนโทรลเลอร์ตระกูล AVR ATmega1284p 8 บิต ทำงานด้วยสัญญาณนาฬิกาความถี่ 16 เมกะเฮิรตซ์ ไมโครคอนโทรลเลอร์ดังกล่าวนี้มีชุดคำสั่งที่มีประสิทธิภาพสูงทั้งหมด 131 คำสั่ง สามารถทำงานแต่ละคำสั่งได้โดยใช้สัญญาณนาฬิกาน้อยสุดเพียง 1 คาบ หน่วยความจำแบบแฟลชขนาด 128 กิโลไบต์ แรมขนาด 16

กิโลไบต์ หน่วยความจำประเภท EEPROM ขนาด 4 กิโลไบต์ และสามารถใช้งานเรจิสเตอร์พิเศษเพื่อป้องกันการดาวน์โหลดโปรแกรม สามารถทำงานได้ในช่วงแรงดันไฟฟ้ากว้าง 1.8 – 5.5 โวลต์ ไมโครคอนโทรลเลอร์ดังกล่าวนี้มีคุณสมบัติการใช้พลังงานต่ำ

การเชื่อมต่อมีช่องต่อประเภทดิจิทัลทั้งหมดจำนวน 32 ช่อง พอร์ตอนุกรม 2 ช่อง บัสเอสพีไอ และ บัส I^2C รองรับสัญญาณในระดับ 3.3 โวลต์ และ 5 โวลต์ นอกจากนี้ยังมีวงจรพิเศษ ได้แก่ วงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลความละเอียด 10 บิต จำนวน 8 ช่อง วงจรนับสัญญาณนาฬิกาขนาด 8 บิตและขนาด 16 บิตทั้งจากสัญญาณภายในและภายนอก วงจรสร้างสัญญาณพัลส์ส่งออกได้จำนวน 6 ช่อง และ วงจรนับสัญญาณนาฬิกา Watchdog สำหรับการตรวจสอบการทำงานของโปรแกรม

3.3.2 วงจรฐานเวลาจริง

การอ้างอิงเวลาจริงของโนดเซนเซอร์เลือกใช้วงจรสำเร็จรูป DS3232 มีความแม่นยำในการทำงานสูงและใช้พลังงานต่ำ สามารถตั้งเวลาและวันที่ได้ในรูปแบบ วัน/วันที่/เดือน/ค.ศ. และ เวลาในรูปแบบ 12 หรือ 24 ชั่วโมง ในวงจรจะคำนวณการเปลี่ยนวันที่ให้โดยอัตโนมัติ สามารถตั้งปลุกตามวันและเวลาที่กำหนดด้วยตัวเองโดยไม่ต้องอาศัยการทำสั่งงานจากไมโครคอนโทรลเลอร์และสามารถให้ข้อมูลอุณหภูมิได้อีกด้วย การเชื่อมต่อกับไมโครคอนโทรลเลอร์หลักใช้การสื่อสารผ่านบัส I^2C การแยกวงจรฐานเวลาจริงออกมาจากไมโครคอนโทรลเลอร์หลักมีจุดเด่น คือ วงจรฐานเวลาหลักจะใช้พลังงานที่ต่ำกว่าเมื่อเทียบกับการใช้งานวงจรฐานเวลาจริงภายในไมโครคอนโทรลเลอร์ ผู้ใช้สามารถสั่งงานไมโครคอนโทรลเลอร์เข้าสู่การทำงานในโหมดนอนหลับเพื่อลดการใช้พลังงานได้โดยไม่กระทบต่อการอ้างอิงฐานเวลาจริง

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

3.3.3 ช่องต่อสำหรับแผงวงจรส่วนต่อขยาย

ช่องต่อสำหรับแผงวงจรส่วนขยายออกแบบให้สามารถเข้าถึงช่องเชื่อมต่อของไมโครคอนโทรลเลอร์โดยตรง สามารถใช้งานช่องเชื่อมต่อดิจิทัลเพื่อใช้สำหรับรับและส่งสัญญาณได้ทั้งหมด 32 ช่อง หรือ กำหนดให้ช่องต่อบางช่องทำหน้าที่เป็นช่องต่อประสานประเภทอื่นได้ เช่น พอร์ตอนุกรม บัสเอสพีไอ บัส I^2C นอกจากนี้ยังมีช่องจ่ายพลังงานไฟฟ้าขนาด 5 โวลต์จากบอร์ดหลักด้วย

3.3.4 วงจรรวมแปลงระดับแรงดันไฟฟ้า

ต้นแบบโนดเซนเซอร์ออกแบบให้ใช้พลังงานได้จากแบตเตอรี่ขนาด 1 เซลล์ที่มีแรงดันไฟฟ้าระหว่าง 3.7 – 4.5 โวลต์ ซึ่งต่างจากบอร์ดอาดูอิโนมาตรฐานที่ออกแบบให้ใช้แรงดันไฟฟ้าขาเข้าขนาด 7 – 12 โวลต์ เพื่อให้ได้ความต่างศักย์ที่สูงขึ้นสำหรับใช้งานกับอุปกรณ์ที่ติดตั้งบนโนดเซนเซอร์ การ

ออกแบบจึงนำวงจรรวม MAX1674 ทำหน้าที่เพิ่มแรงดันไฟฟ้าแระแสตรง (step-up DC-DC converter) จากแบตเตอรี่ขนาด 3.7 โวลต์ ให้มีขนาดสูงขึ้นเท่ากับ 5 โวลต์ วงจรรวมสามารถจ่ายกระแสได้สูงสุด 1 แอมแปร์ วงจรรวมนี้ยังสามารถทำงานได้ปกติแม้แรงดันด้านขาเข้าวงจรรวมจะมีขนาดเหลือเพียง 1.8 โวลต์ ก็ตาม การใช้วงจรแปลงระดับแรงดันจะช่วยให้ไมโครคอนโทรลเลอร์และอุปกรณ์ที่นำมาเชื่อมต่อทำงานได้อย่างมีเสถียรภาพ โดยไม่ได้รับผลกระทบจากขนาดของแรงดันไฟฟ้าที่เปลี่ยนแปลงเมื่อแบตเตอรี่มีคายประจุในระหว่างการใช้งาน

3.3.5 วงจรรวมตรวจวัดแบตเตอรี่

ข้อมูลพลังงานคงเหลือภายในแบตเตอรี่ของโนดเซนเซอร์เป็นข้อมูลที่มีความสำคัญถูกนำใช้เป็นข้อมูลเพื่อใช้ในการออกแบบพัฒนารูปแบบการทำงานและยืดระยะเวลาการทำงานให้ได้ระยะเวลานานที่สุดต่อรอบการเปลี่ยนแบตเตอรี่ ตัวอย่างเช่น การออกแบบการทำงานให้มีการตรวจวัดที่มีความถี่ลดลงเมื่อพลังงานภายในแบตเตอรี่เหลือน้อย หรือ การเปลี่ยนเส้นทางการสื่อสารให้ใช้เส้นทางที่ไม่ผ่านโนดที่มีพลังงานเหลือน้อย เป็นต้น การออกแบบการตรวจวัดพลังงานที่เหลืออยู่ภายในแบตเตอรี่จึงมีความสำคัญซึ่งบอร์ดอาดูอิโนมาตรฐานทุกรุ่นไม่ได้มีการติดตั้งไว้ให้ใช้งาน โดยเลือกใช้วงจรรวมสำหรับตรวจวัดแบตเตอรี่ MAX17043 จากบริษัท MAXIM ทำการตรวจวัดแบตเตอรี่ขนาด 1 เซลล์ ให้ข้อมูลแรงดันไฟฟ้าภายในแบตเตอรี่มีหน่วยเป็นโวลต์ และ ข้อมูลความจุไฟฟ้าคงเหลือภายในแบตเตอรี่มีหน่วยเป็นเปอร์เซ็นต์ จุดเด่นของวงจรรวมตัวดังกล่าวนี้ คือ ไม่ต้องการการต่อตัวต้านทาน (sense resistor) จากภายนอก และ ไม่ต้องการขั้นตอนการเรียนรู้การใช้งานแบตเตอรี่ก่อน (full-to-empty battery relearning) การตรวจวัดสามารถยังสามารถตั้งเตือนโดยใช้สัญญาณอินเทอร์รัพท์ไปยังไมโครคอนโทรลเลอร์เมื่อความจุของแบตเตอรี่เหลือน้อยกว่าเกณฑ์ที่กำหนดไว้ การเชื่อมต่อกับไมโครคอนโทรลเลอร์ใช้การเชื่อมต่อด้วยบัส I^2C

3.4 โครงสร้างบอร์ดส่วนต่อขยาย

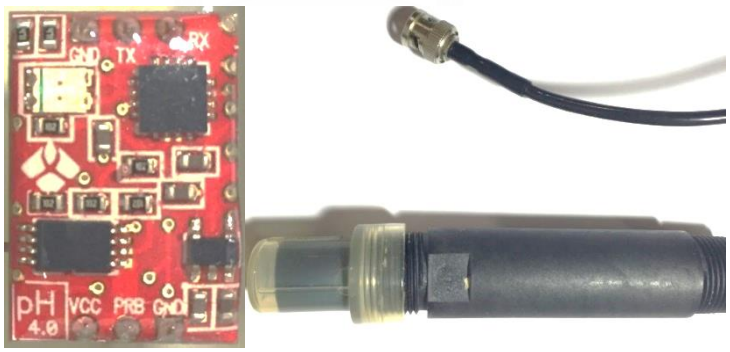
การออกแบบวงจรบอร์ดส่วนต่อขยายทำหน้าที่เป็นส่วนที่ใช้ติดตั้งของส่วนตรวจวัดและสั่งการ และ ส่วนสื่อสารไร้สาย โดยออกแบบเป็นช่องต่อบนแผงวงจรสามารถนำอุปกรณ์เซนเซอร์ เครื่องรับส่งสัญญาณวิทยุไร้สาย หรือ อุปกรณ์ที่นำต่อพ่วงอื่น ๆ มาติดตั้งได้ แผงวงจรส่วนต่อขยายจะติดตั้งซ้อนด้านบนของแผงวงจรหลัก การรับส่งข้อมูลของอุปกรณ์ที่นำมาติดตั้งจะผ่านจุดเชื่อมต่อระหว่างแผงวงจรส่วนต่อขยายและวงจรหลัก บอร์ดส่วนต่อขยายมีรายละเอียดของส่วนประกอบของอุปกรณ์ต่าง ๆ ดังต่อไปนี้

3.4.1 อุปกรณ์เซนเซอร์

การเชื่อมต่อกันระหว่างไมโครคอนโทรเลอร์และเซนเซอร์มีรูปแบบการเชื่อมต่อได้ 2 ประเภท คือ การเชื่อมต่อแบบแอนะล็อก และ การเชื่อมต่อแบบดิจิทัล การออกแบบช่องเชื่อมต่อจึงออกแบบให้ช่องติดตั้งเซนเซอร์บนบอร์ดขยายสามารถเชื่อมต่อเซนเซอร์ได้ทั้ง 2 รูปแบบ ในแต่ละช่องเชื่อมต่ออุปกรณ์เซนเซอร์นั้นภายในออกแบบให้มีช่องเชื่อมต่อประเภทดิจิทัล ช่องเชื่อมต่อประเภทแอนะล็อก พอร์ตอนุกรม และ บัส I^2C รวมถึงช่องสำหรับจ่ายพลังงานไฟฟ้าขนาด 5 โวลต์ และ 3.3 โวลต์ สามารถสั่งเปิดและปิดการจ่ายพลังงานโดยผ่านทรานซิสเตอร์ที่ทำหน้าที่เป็นสวิตช์ ในงานวิจัยนี้จะเลือกนำตัวอย่างเซนเซอร์ที่มาติดตั้งสำหรับงานตรวจวัดคุณภาพแหล่งน้ำดังต่อไปนี้

3.4.1.1 เซนเซอร์ตรวจวัดค่าพีเอช

เซนเซอร์ตรวจวัดค่าพีเอช (pH sensor) ประเภทดิจิทัล pH Circuit รุ่น 4.0 พร้อมหัววัด (probe) จากบริษัท AtlasScientific สำหรับตรวจวัดความเป็นกรด-ด่างของสายละลาย มีจุดเด่นในการใช้งานเรื่องการตรวจวัดและความง่ายในการเชื่อมต่อ ค่าที่ตรวจวัดเป็นระดับพีเอชในช่วง 0.01-14.00 ความละเอียดในระดับทศนิยมสองตำแหน่ง และ สามารถกำหนดค่าอุณหภูมิของสารละลายเพื่อเป็นพารามิเตอร์เพิ่มในการตรวจวัด การปรับค่าให้เป็นมาตรฐาน (calibration) สามารถทำได้โดยใช้คำสั่งเพียงคำสั่งเดียว การเชื่อมต่อกับหน่วยประมวลผลผ่านพอร์ตอนุกรมโดยใช้รหัสแอสกี (ASCII code) เป็นคำสั่งในการใช้งานและผลลัทธ์ในการตรวจวัด มีหลอดแอลอีดี (LED) สำหรับแสดงสถานะการทำงาน ขณะใช้งานใช้กระแสไฟฟ้าขนาด 2 มิลลิแอมแปร์ แรงดันไฟฟ้าขนาด 3.3 โวลต์



รูปที่ 3.6 เซนเซอร์ pH Circuit รุ่น 4.0 พร้อมหัววัด

3.4.1.2 เซนเซอร์ตรวจวัดการละลายออกซิเจนในน้ำ

เซนเซอร์ตรวจวัดการละลายออกซิเจนในน้ำ (D.O. sensor) ประเภทดิจิทัล D.O. Circuit รุ่น 6.0 พร้อมหัววัด จากบริษัท AtlasScientific สำหรับตรวจวัดการละลายของออกซิเจนในน้ำในหน่วยมิลลิกรัมต่อลิตร (mg/L) โดยมีความละเอียดในการวัดในระดับทศนิยมสามตำแหน่ง ชุดเซนเซอร์นี้มี

จุดเด่นในการนำมาใช้งานโดยสามารถวัดออกซิเจนที่ละลายได้ในน้ำจืด น้ำกร่อย และ น้ำเค็ม ไม่ต้องการวงจรภายนอกสำหรับการปรับค่าให้เป็นมาตรฐานสามารถซึ่งทำได้โดยใช้เพียง 1 คำสั่ง การเชื่อมต่อกับหน่วยประมวลผลผ่านพอร์ตอนุกรมโดยใช้รหัสแอสกี้เป็นคำสั่งในการใช้งานและผลลัทธ์ในการตรวจวัด มีหลอดแอลอีดีสำหรับแสดงสถานะการทำงาน ขณะใช้งานใช้กระแสไฟฟ้าขนาด 4.7 มิลลิแอมแปร์ แรงดันไฟฟ้าขนาด 3.3 โวลต์

รูปที่ 3.7 เซนเซอร์ D.O. Circuit รุ่น 6.0 พร้อมหัววัด

3.4.1.3 เซนเซอร์ตรวจวัดอุณหภูมิ

เซนเซอร์ตรวจวัดอุณหภูมิ (temperature sensor) ประเภทดิจิทัล DS18B20 จากบริษัท Dallas Semiconductor ตรวจวัดอุณหภูมิได้ในช่วง 55 – 125 องศาเซลเซียส สามารถกำหนดค่าความละเอียดในการวัดในระดับทศนิยมสามตำแหน่ง ใช้เวลาในการตรวจวัดสูงสุดไม่เกิน 750 มิลลิวินาทีต่อครั้ง การเชื่อมต่อกับหน่วยประมวลผลผ่านบัส OneWire ซึ่งเป็นจุดเด่นในการเชื่อมต่อโดยใช้สายสัญญาณข้อมูลเพียง 1 เส้น สามารถทำงานที่ระดับแรงดันไฟฟ้า 3 – 5.5 โวลต์ ข้อมูลอุณหภูมิจะเก็บอยู่ในเรจิสเตอร์ขนาด 8 บิต จำนวน 2 ตัว การเข้าถึงเรจิสเตอร์ที่เก็บค่าอุณหภูมิจะใช้รูปแบบวิธีสำหรับอุปกรณ์ที่เชื่อมต่อผ่านบัส OneWire ค่าที่อ่านได้จากเรจิสเตอร์จะต้องนำมาคำนวณกลับภายในโปรแกรมเพื่อแปลงเป็นข้อมูลอุณหภูมิอีกครั้ง

รูปที่ 3.8 เซนเซอร์ตรวจวัดอุณหภูมิ DS18B20

3.4.2 เครื่องรับส่งสัญญาณวิทยุ

การเชื่อมต่อเครื่องรับส่งสัญญาณวิทยุที่มีจำหน่ายในท้องตลาดทั่วไปมีรูปแบบการเชื่อมต่อหลัก 2 ประเภท คือ การเชื่อมต่อผ่านพอร์ตอนุกรมและการเชื่อมต่อผ่านบัสเอสพีไอ การออกแบบช่องเชื่อมต่อเครื่องรับส่งสัญญาณวิทยุสำหรับต้นแบบโนดเซนเซอร์จึงออกแบบให้ช่องเชื่อมต่อเครื่องรับส่งสัญญาณวิทยุประกอบด้วยช่องเชื่อมต่อดิจิทัล ช่องเชื่อมต่ออนุกรม และ บัสเอสพีไอ รวมถึงช่องสำหรับจ่ายพลังงานไฟฟ้าขนาด 3.3 โวลต์ เพื่อเป็นการแก้ปัญหาการขาดช่องเชื่อมต่อของโนดเซนเซอร์ที่มีจำหน่ายและการปรับเปลี่ยนไปใช้เครื่องรับส่งสัญญาณวิทยุประเภทอื่นได้สะดวกโดยไม่ทำให้ต้นทุนของอุปกรณ์สูงขึ้นเมื่อเทียบกับโนดเซนเซอร์ประเภทที่ติดตั้งเครื่องรับส่งสัญญาณวิทยุแบบถาวร ต้นแบบโนดเซนเซอร์ในงานวิจัยนี้สามารถเลือกติดตั้งเครื่องรับส่งสัญญาณวิทยุได้ครั้งละ 1 เครื่อง โดยตัวอย่างเครื่องรับส่งสัญญาณวิทยุที่เลือกใช้งานมีทั้งหมด 3 ประเภท ได้แก่ เครื่องรับส่งสัญญาณวิทยุที่ไม่มีโพรโทคอล เครื่องรับส่งสัญญาณวิทยุที่ใช้โพรโทคอลซิกบี และ เครื่องรับส่งสัญญาณวิทยุบูลทูท โดยมีจุดประสงค์เพื่อให้รองรับความต้องการของผู้ใช้งานในด้านการเลือกใช้โพรโทคอลการสื่อสารที่เป็นมาตรฐานหรือการพัฒนาโพรโทคอลขึ้นมาใช้งานเอง การเชื่อมต่อกับอุปกรณ์โทรศัพท์สมาร์ตโฟน (smartphone) หรือ แท็บเล็ต (tablet) ตลอดจนลดค่าใช้จ่ายของเครื่องรับส่งสัญญาณวิทยุที่มีราคาต่างกันให้เหมาะสมกับงานที่นำไปประยุต์ใช้ รายละเอียดเครื่องรับส่งสัญญาณแต่ละประเภทที่เลือกนำมาใช้งานมีดังต่อไปนี้

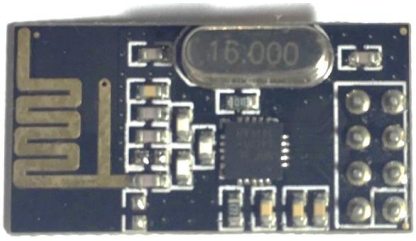
3.4.2.1 เครื่องรับส่งสัญญาณวิทยุ Xbee PRO Zigbee S2

เครื่องรับส่งสัญญาณวิทยุ Xbee PRO Zigbee S2 จากบริษัท Digi เป็นเครื่องรับส่งสัญญาณวิทยุที่ทำงานบนย่านความถี่ 2.4 กิกกะเฮิรตซ์ ใช้โพรโทคอลซิกบีร่วมกับมาตรฐาน IEEE 802.15.4 สำหรับเครือข่ายที่ใช้พลังงานต่ำ ครอบคลุมการสื่อสารในระดับชั้น Physical layer จนถึงระดับชั้น Application layer เครื่องรับส่งสัญญาณวิทยุ Xbee PRO Zigbee S2 สามารถกำหนดรูปแบบการทำงานเพื่อทำหน้าที่เป็นโนดทั้ง 3 ประเภท ได้แก่ Coordinator Router และ End-device ได้ตามข้อกำหนดของเครือข่ายซิกบี สามารถทำได้โดยการเปลี่ยนเฟิร์มแวร์ (firmware) ภายในของเครื่องรับส่งสัญญาณวิทยุ หรือ เปลี่ยนรูปแบบเครือข่ายไปเป็นรูปแบบตาข่าย (mesh) ที่มีความซับซ้อนที่ใช้พื้นฐานการพัฒนาจาก XBee ZB ZigBee mesh ความเร็วในการรับส่งข้อมูลสูงสุด 250 กิโลบิตต่อวินาที การเชื่อมต่อกับไมโครคอนโทรลเลอร์จะเชื่อมต่อผ่านพอร์ตอนุกรม การใช้งานสามารถใช้คำสั่ง AT command หรือ การกลุ่มของคำสั่งที่มีรูปแบบที่กำหนดไว้ ใช้พลังงานไฟฟ้าแรงดัน 3.3 โวลต์ ราคาจำหน่ายประมาณสองพันบาทต่อเครื่อง

รูปที่ 3.9 เครื่องรับส่งสัญญาณวิทยุ Xbee PRO Zigbee S2

3.4.2.2 เครื่องรับส่งสัญญาณวิทยุ nRF24L01+

เครื่องรับส่งสัญญาณวิทยุ nRF24L01+ จากบริษัท Nordic เป็นเครื่องรับส่งสัญญาณวิทยุที่ทำงานบนย่านความถี่ 2.400 – 2.525 กิกกะเฮิรตซ์ มอดูเลต (modulate) ชนิด GFSK สามารถเลือกช่องสัญญาณได้ทั้งหมด 25 ช่อง โดยกำหนดให้มีความกว้างช่องสัญญาณ 5 เฮิรตซ์ ความเร็วสูงสุดในการส่งข้อมูล 2 เมกกะบิตต่อวินาที สามารถปรับระดับกำลังในการส่งข้อมูลได้ ไม่มีโพรโทคอลในชั้น Data link layer ผู้ใช้งานสามารถพัฒนาโพรโทคอลการสื่อสารขึ้นมาใช้งานได้เอง การเชื่อมต่อกับไมโครคอนโทรลเลอร์ผ่านบัสเอสพีไอและช่องเชื่อมต่อดิจิทัลสำหรับการควบคุมโหมดการทำงานรับส่งข้อมูล ใช้พลังงานส่งข้อมูล 11.3 มิลลิแอมแปร์ในการส่งข้อมูลที่กำลัง 0 dBm สามารถทำงานที่ความต่างศักย์ระหว่าง 1.9 – 3.3 โวลต์ พร้อมวงจรเพื่อรักษาแรงดันไฟฟ้าในการทำงาน และ รองรับระดับสัญญาณขนาด 5 โวลต์ ในบัสเอสพีไอด้วย เครื่องรับส่งสัญญาณวิทยุ nRF24L01+ มีราคาจำหน่ายประมาณหนึ่งร้อยบาทต่อเครื่อง

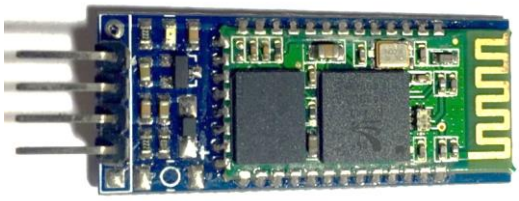


รูปที่ 3.10 เครื่องรับส่งสัญญาณวิทยุ nRF24L01+

3.4.2.3 เครื่องรับส่งสัญญาณวิทยุ HC-05

เครื่องรับส่งสัญญาณวิทยุ HC-05 ใช้เทคโนโลยีบูลทูท (Bluetooth) รุ่น 2.0 การทำงานจะเป็นการรับส่งข้อมูลของคู่อุปกรณ์โดยตัวเครื่องรับส่งสัญญาณวิทยุ HC-05 จะทำงานในโหมดสเลฟ (slave mode) อุปกรณ์ที่จะนำมาเชื่อมต่อจะต้องมำงานในโหมดมาสเตอร์ (master mode) เช่น

คอมพิวเตอร์ โทรศัพท์มือ หรือ อุปกรณ์อื่น ๆ และ ทำการจับคู่อุปกรณ์ (paring) ไว้ก่อนล่วงหน้าถึงจะทำการส่งข้อมูลผ่านคลื่นวิทยุได้ การเชื่อมต่อกับไมโครคอนโทรลเลอร์ใช้การเชื่อมต่อผ่านพอร์ตอนุกรม รับส่งข้อมูลในรูปแบบรหัสแอสกี สามารถตั้งค่าการทำงานด้วยคำสั่ง AT Command ก่อนทำการจับคู่อุปกรณ์ รูปแบบคำสั่งการกำหนดค่าคล้ายกับการสั่งงานโมเด็ม (modem) ทั่วไป ราคาจำหน่ายประมาณสองร้อยบาทต่อเครื่อง



รูปที่ 3.11 เครื่องรับส่งสัญญาณวิทยุ HC-05

3.5 การเชื่อมต่อกับคอมพิวเตอร์

การเชื่อมต่อระหว่างโนดเซนเซอร์กับคอมพิวเตอร์ ในขั้นตอนการพัฒนาหรือทดสอบการทำงานอาจแบ่งประเภทของการเชื่อมต่อตามลักษณะการต่อเชื่อมได้ 2 ประเภท ดังนี้

3.5.1 การเชื่อมต่อโดยใช้เครื่องโปรแกรม

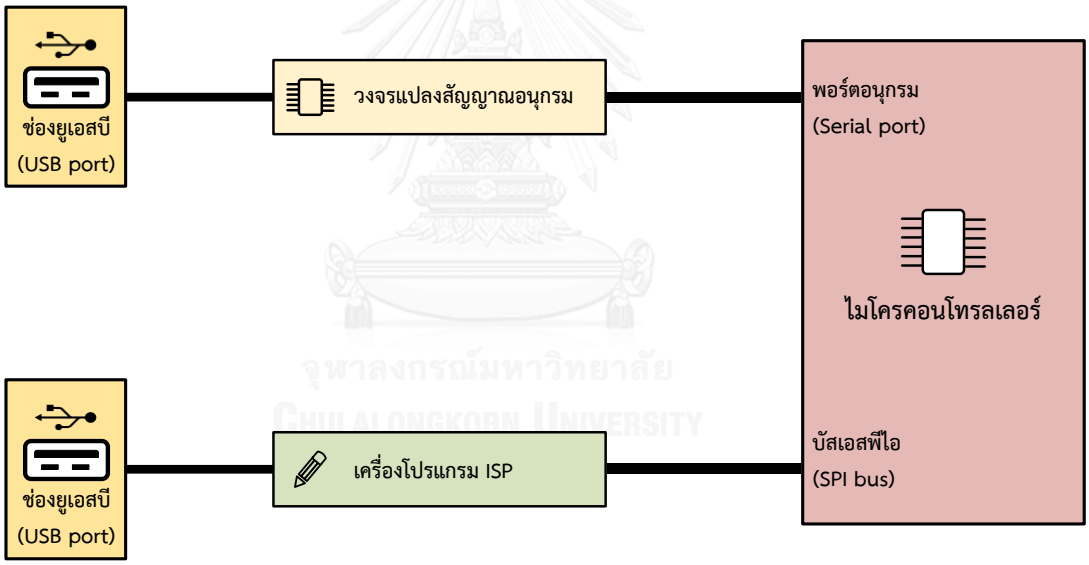
การเชื่อมต่อด้วยเครื่องโปรแกรม (programmer) มีจุดประสงค์เพื่อใช้เป็นช่องทางเพื่อทำการดาวน์โหลดโปรแกรมลงในหน่วยความจำแบบแฟลชของไมโครคอนโทรลเลอร์โดยตรง สามารถใช้กำหนดค่าเรจิสเตอร์พิเศษของไมโครคอนโทรลเลอร์ได้ เครื่องโปรแกรมโปรแกรม ISP (in-system programming) หรือ ICSP (in-circuit serial programming) จะใช้ช่องเชื่อมต่อร่วมกับช่องสัญญาณของบัสเอสพีไอ การดาวน์โหลดโปรแกรมยังสามารถใช้เครื่องดีบักเกอร์ (debugger) ได้เช่นกัน เมื่อทำการดาวน์โหลดโปรแกรมด้วยเครื่องดีบักเกอร์เสร็จ สามารถใช้ซอฟต์แวร์พิเศษบนคอมพิวเตอร์เพื่อทดสอบการทำงานของโปรแกรมได้ด้วยการตั้งจุดหยุดชั่วคราว (break point) ที่รหัสต้นฉบับภายในโปรแกรมบนคอมพิวเตอร์แล้วตรวจสอบค่าตัวแปรหรือเรจิสเตอร์ได้อีกด้วย

3.5.2 การเชื่อมต่อโดยใช้พอร์ตอนุกรม

การเชื่อมต่อโดยใช้พอร์ตอนุกรมมีจุดประสงค์เพื่อสื่อสารกับคอมพิวเตอร์ แสดงผลการทำงานของโปรแกรม การรับส่งข้อมูลจากคอมพิวเตอร์เพื่อเป็นทางเชื่อมต่อกับเครือข่ายอื่น ในอดีตการเชื่อมต่อดังกล่าวจะใช้พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ผ่านไปยังชุดวงจรแปลงระดับสัญญาณมาตรฐาน RS-232 และต่อไปยังพอร์ตอนุกรมของคอมพิวเตอร์ แต่ในปัจจุบันคอมพิวเตอร์ส่วนใหญ่ไม่

ติดตั้งพอร์ตอนุกรมให้ใช้งานแล้ว คอมพิวเตอร์จะเชื่อมต่อกับอุปกรณ์ภายนอกผ่านพอร์ตยูเอสบีเป็นส่วนใหญ่

การออกแบบการเชื่อมต่อกับคอมพิวเตอร์สำหรับต้นแบบโนดเซนเซอร์จึงออกแบบให้สามารถเชื่อมกับคอมพิวเตอร์ได้ทั้งการใช้งานผ่านเครื่องโปรแกรมประเภท ISP และการเชื่อมต่อโดยใช้พอร์ตอนุกรมโดยเพิ่มวงจรรวม FTDI 232 แปลงรูปแบบการสื่อสารด้วยพอร์ตอนุกรมให้อยู่ในรูปแบบการสื่อสารด้วยช่องยูเอสบีเพื่ออำนวยความสะดวกกับผู้ใช้งานทั่วไป การใช้งานบนคอมพิวเตอร์รุ่นใหม่จะต้องติดตั้งโปรแกรมขับอุปกรณ์เพิ่ม เพื่อให้ระบบปฏิบัติการสามารถเรียกใช้งานพอร์ตอนุกรมเสมือน (virtual serial port) ได้ ภาพรวมการเชื่อมต่อระหว่างโนดเซนเซอร์และคอมพิวเตอร์แสดงในรูปที่ 3.12 การออกแบบให้สามารถเชื่อมต่อคอมพิวเตอร์ได้ทั้ง 2 รูปแบบทำให้ผู้ใช้มีทางเลือกดาวน์โหลดโปรแกรมได้สะดวกมากขึ้น ตลอดจนใช้งานการแสดงผลผ่านพอร์ตอนุกรมเพื่อแสดงผลการทำงาน หรือตรวจสอบข้อผิดพลาดการทำงานได้



รูปที่ 3.12 การเชื่อมต่อระหว่างคอมพิวเตอร์และโนดเซนเซอร์

3.6 แหล่งพลังงานของโนดเซนเซอร์

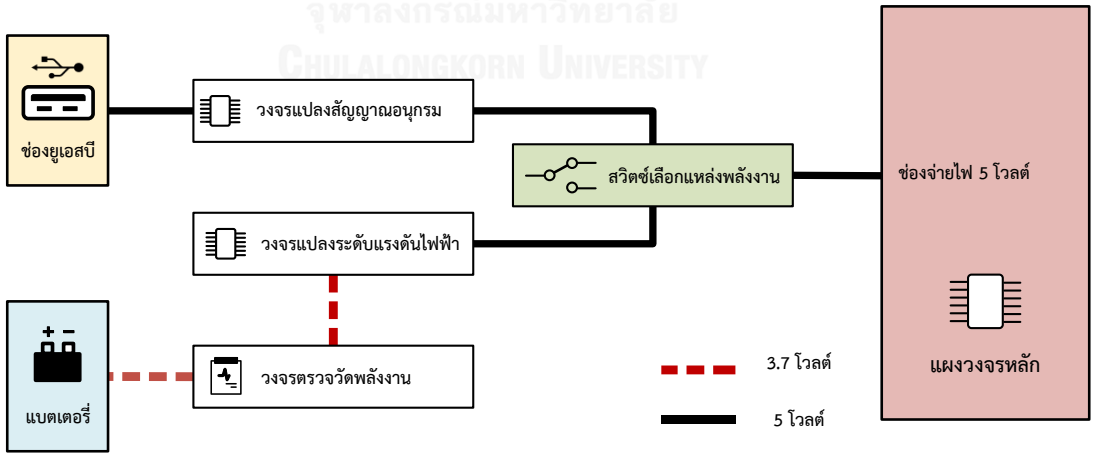
การใช้พลังงานของโนดเซนเซอร์ใช้พลังงานไฟฟ้าขนาด 5 โวลต์สำหรับการทำงานของไมโครคอนโทรลเลอร์และอุปกรณ์บนแผงวงจรหลัก ในส่วนแผงวงจรส่วนขยายจะรับพลังงานไฟฟ้าจากแผงวงจรหลักและติดตั้งวงจรรักษาแรงดันขนาด 3.3 โวลต์ ใช้สำหรับเซนเซอร์บางประเภทรวมถึงเครื่องรับส่งสัญญาณวิทยุ พลังงานของโนดเซนเซอร์ถูกออกแบบให้สามารถเลือกใช้พลังงานได้จากแหล่งที่มา 2 แหล่ง ดังนี้

3.6.1 ช่องยูเอสบี

การเชื่อมต่อผ่านช่องยูเอสบี 2.0 มีการกำหนดมาตรฐานสำหรับการจ่ายพลังงานไฟฟ้าให้กับอุปกรณ์ที่มาต่อพ่วงได้ โดยใช้ความต่างศักย์ 5 โวลต์ กระแสไฟฟ้าขนาดไม่เกิน 500 มิลลิแอมแปร์ การออกแบบให้ใช้พลังงานไฟฟ้าได้จากช่องยูเอสบีมีความสะดวกสำหรับผู้พัฒนาในขั้นตอนการพัฒนาโปรแกรมบนคอมพิวเตอร์ การเชื่อมต่อสามารถใช้ช่องยูเอสบีเพียงช่องเดียวเพื่อทำการดาวน์โหลดโปรแกรมลงบนโนดเซนเซอร์ ตรวจสอบข้อผิดพลาดการทำงาน และใช้เป็นแหล่งพลังงานโดยไม่จำเป็นต้องต่อแบตเตอรี่จากภายนอกเพิ่มเติม

3.6.2 แบตเตอรี่

การใช้พลังงานจากแบตเตอรี่ออกแบบให้สามารถใช้พลังงานไฟฟ้าที่มีแรงดันขนาด 1.8-5.5 โวลต์ ซึ่งจะผ่านวงจรเพิ่มระดับแรงดันไฟฟ้าให้มีขนาด 5 โวลต์ ก่อนนำไปใช้งานภายในโนดเซนเซอร์ ในการทดลองนี้จะเลือกใช้แบตเตอรี่ลิเธียม-ไอออน (Lithium-Ion) จำนวน 1 เซลล์ ความจุ 2,000 มิลลิแอมแปร์ชั่วโมง 3.7 โวลต์ สามารถประจุไฟซ้ำได้ การเลือกใช้แบตเตอรี่ดังกล่าวเนื่องจากคุณลักษณะของแบตเตอรี่ที่มีขนาดเล็ก น้ำหนักเบา สามารถจ่ายพลังงานได้อย่างต่อเนื่องโดยความต่างศักย์ของแบตเตอรี่ค่อนข้างคงที่ไม่ลดลงเมื่อแบตเตอรี่ทำการจ่ายไฟออก นอกจากนี้ยังสามารถประจุไฟซ้ำได้โดยไม่ต้องรอให้แบตเตอรี่หมดประจุไฟและไม่มีผลกระทบจากหน่วยความจำของแบตเตอรี่ (memory effect) ในการประจุไฟแต่ละครั้งอีกด้วย สำหรับการพัฒนาให้สามารถประจุไฟกลับโดยใช้แหล่งพลังงานภายนอกอื่น สามารถติดตั้งวงจรรวมสำหรับประจุไฟใหม่ได้สะดวก



รูปที่ 3.13 แหล่งพลังงานของโนดเซนเซอร์

บทที่ 4
การพัฒนาส่วนต่อประสานโปรแกรมประยุกต์

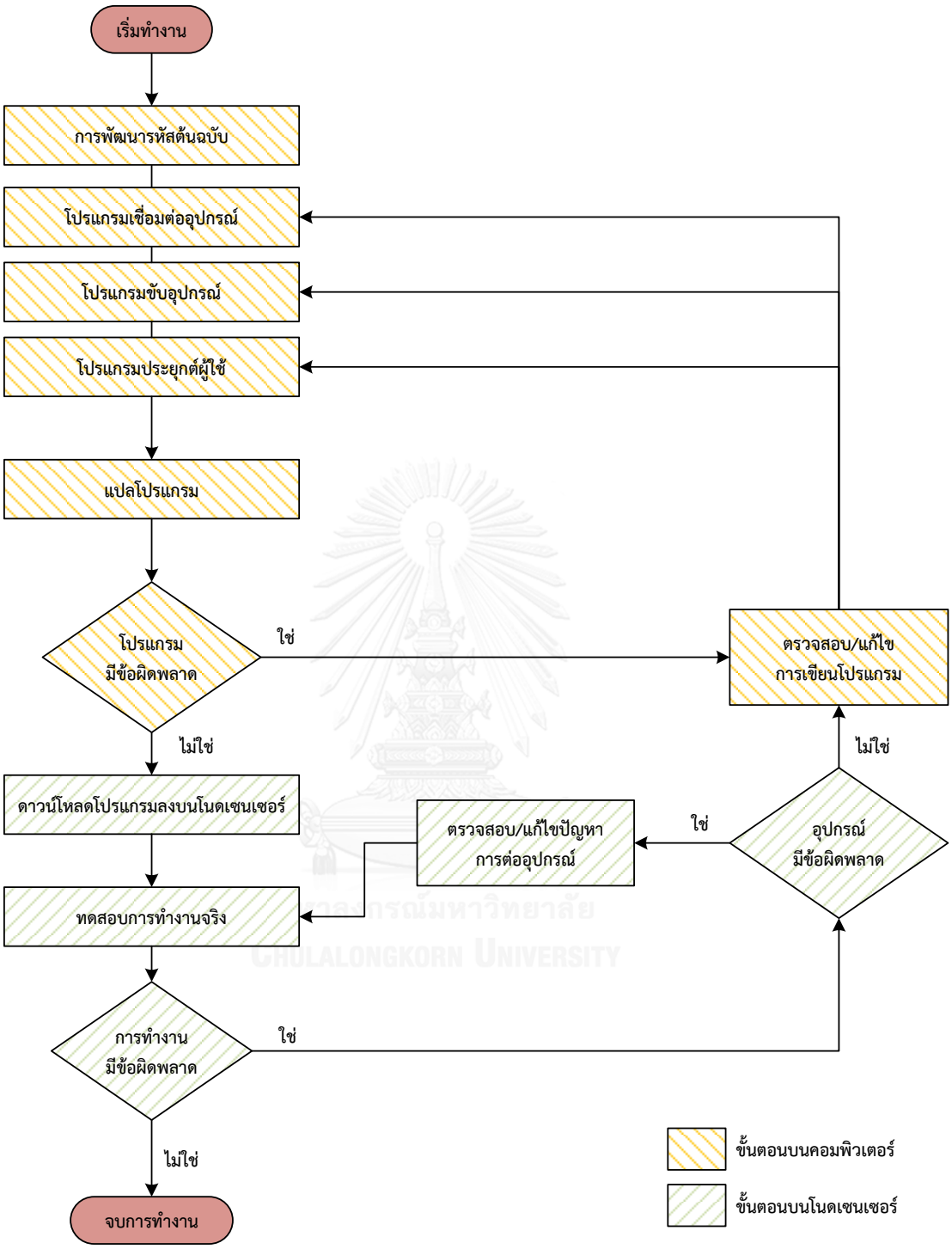
4.1 ภาพรวมการพัฒนาโปรแกรมประยุกต์สำหรับโนดเซนเซอร์

การพัฒนาโปรแกรมประยุกต์สำหรับโนดเซนเซอร์มีขั้นตอนที่สำคัญในการพัฒนาแบ่งได้เป็น 2 ช่วงดังนี้

ช่วงที่ 1 การพัฒนาบนคอมพิวเตอร์ การพัฒนารหัสต้นฉบับสำหรับโปรแกรมประยุกต์สามารถเลือกใช้โปรแกรมแก้ไขรหัสต้นฉบับ (code editor) อาทิ โปรแกรม Notepad, Vim, EditPlus แต่ผู้ใช้อาจเลือกใช้โปรแกรมประเภท IDE (integrated development environment) ซึ่งรวบรวมเครื่องมือที่จำเป็นในการพัฒนาไว้ทั้งหมดภายในที่เดียว เช่น โปรแกรม Atmel Studio, CodeWarrior, ArduinoIDE ผู้ใช้อาจจะต้องสร้างโปรแกรมขับอุปกรณ์สำหรับสั่งงานอุปกรณ์ขึ้นมาใช้เอง ซึ่งใช้ระยะเวลาในการพัฒนาค่อนข้างมากสำหรับผู้เริ่มต้นใช้งานเป็นครั้งแรกหรือผู้ที่ไม่คุ้นชินกับการต่อวงจรอิเล็กทรอนิกส์ หลังจากการพัฒนารหัสต้นฉบับเสร็จต้องทำการแปลโปรแกรม (compile) เพื่อให้ได้โปรแกรมคำสั่งภาษาเครื่อง (machine language) เพื่อนำไปใช้ในขั้นตอนถัดไป

ช่วงที่ 2 การทดสอบบนอุปกรณ์จริง เมื่อได้ไฟล์ภาษาเครื่องแล้วนั้นจะทำการดาวน์โหลดโปรแกรมลงบนหน่วยความจำของโนดเซนเซอร์ การทดสอบการทำงานโปรแกรมประยุกต์จะต่อโนดเซนเซอร์กับอุปกรณ์จริงร่วมกับเครื่องมือที่ใช้ตรวจสอบ เช่น มัลติมิเตอร์ (multi-meter) ออสซิลโลสโคป (oscilloscope) และเครื่องมือวิเคราะห์สัญญาณลอจิก (logic analyzer)

เมื่อตรวจพบข้อผิดพลาดของการทำงาน การแก้ไขปัญหาจะต้องระบุสาเหตุของข้อผิดพลาดก่อนทำการแก้ไขซึ่ง ข้อผิดพลาดอาจมาจากปัญหาทางฮาร์ดแวร์หรือซอฟต์แวร์ก็เป็นได้ การระบุสาเหตุข้อผิดพลาดอาจเริ่มจากการตรวจสอบอุปกรณ์ เช่น การต่อสายไฟระหว่างอุปกรณ์ที่ไม่ถูกต้อง ปัญหาการเชื่อมต่ออุปกรณ์ที่ไม่แน่นสนิทหรือปัญหาการจ่ายพลังงานไฟฟ้าที่ไม่ถูกต้องแล้วทดสอบการทำงานอีกครั้งหากยังพบข้อผิดพลาดดังเดิม ปัญหาที่เกิดขึ้นอาจมาจากการเขียนชุดคำสั่งของโปรแกรมไม่ถูกต้อง ผู้พัฒนาจะต้องกลับไปแก้ไขโปรแกรมประยุกต์ในช่วงที่ 1 อีกครั้งและทำการทดสอบซ้ำดังแสดงขั้นตอนการพัฒนาในรูปที่ 4.1 จนไม่เกิดข้อผิดพลาดในการทำงาน



รูปที่ 4.1 แผนผังขั้นตอนการพัฒนาซอฟต์แวร์สำหรับโนดเซนเซอร์

4.2 การโปรแกรมลงบนโนดเซนเซอร์

การดาวน์โหลดโปรแกรมที่พัฒนาเสร็จแล้วจากคอมพิวเตอร์ลงบนโนดเซนเซอร์สามารถทำได้ 2 วิธี ดังต่อไปนี้

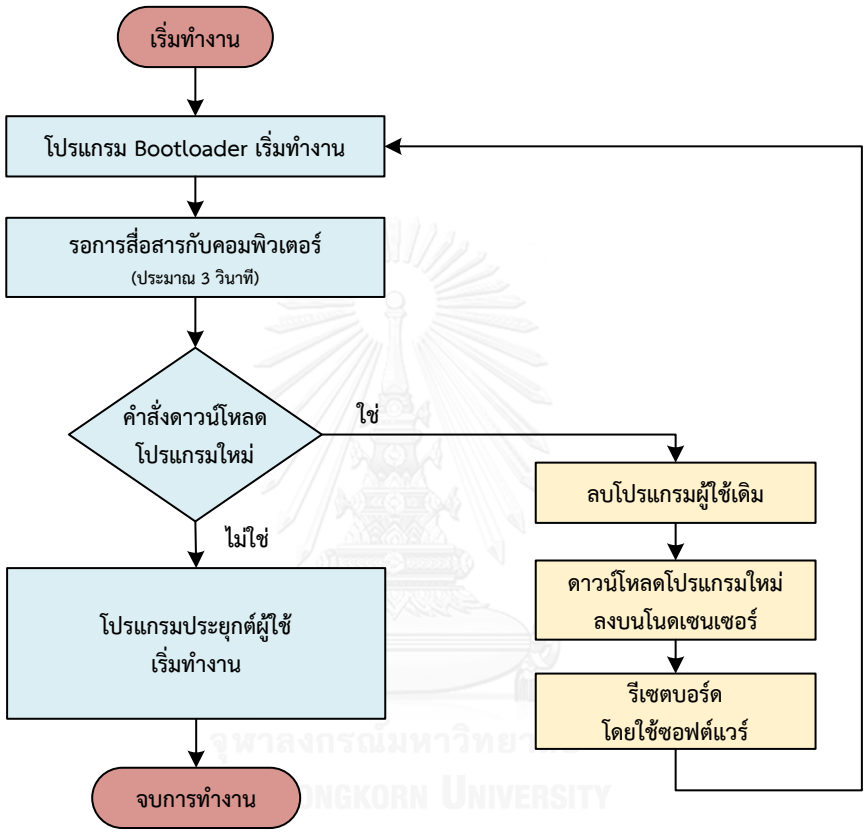
4.2.1 การโปรแกรมโดยใช้เครื่องโปรแกรม

การใช้เครื่องโปรแกรม (programmer) เป็นวิธีพื้นฐานสำหรับดาวน์โหลดโปรแกรมที่แปลเสร็จแล้วลงบนหน่วยความจำของไมโครคอนโทรลเลอร์ทั่วไป รวมถึงไมโครคอนโทรลเลอร์ที่ใช้งานเป็นหน่วยประมวลผลของโนดเซนเซอร์ เครื่องโปรแกรมจะทำหน้าที่เขียนคำสั่งลงในหน่วยความจำของไมโครคอนโทรลเลอร์โดยตรง ผ่านช่องต่อประสานประเภทบัสเอสพีไอ ช่องดีบักเกอร์ หรือ ช่องต่อพิเศษอื่น ๆ ตามที่มีการออกแบบไว้ เครื่องโปรแกรมจะสามารถเข้าไปเขียนหรือลบค่าในหน่วยความจำได้ทั้งหมดโดยไม่ต้องติดตั้งโปรแกรมใด ๆ บนไมโครคอนโทรลเลอร์ รวมถึงการกำหนดค่าเรจิสเตอร์พิเศษสำหรับการใช้งานในครั้งแรกอีกด้วย เครื่องโปรแกรมจะเชื่อมต่อกับคอมพิวเตอร์ผ่านพอร์ตขนาน พอร์ตอนุกรม หรือ พอร์ตยูเอสบี เครื่องโปรแกรมจะมีความเฉพาะเจาะจงสามารถใช้งานกับรุ่นหรือตระกูลของไมโครคอนโทรลเลอร์ที่รองรับเท่านั้น ไม่สามารถใช้งานข้ามรุ่นหรือตระกูลได้ ในอดีตการโปรแกรมด้วยเครื่องโปรแกรมอาจต้องถอดไมโครคอนโทรเลอร์ออกจากบอร์ดเพื่อนำมาติดตั้งบนเครื่องโปรแกรม แต่ในปัจจุบันเครื่องโปรแกรมรุ่นใหม่สามารถดาวน์โหลดโปรแกรมลงบนหน่วยความจำได้โดยไม่ต้องถอดออกมาจากแผงวงจรสะดวกต่อการใช้งานมากขึ้น ลดความเสียหายที่อาจจะเกิดขึ้นกับขาสัญญาณของไมโครคอนโทรลเลอร์ และราคาต่ำลงกว่าเดิม

4.2.2 การโปรแกรมโดยใช้เฟิร์มแวร์

การใช้เฟิร์มแวร์ (firmware) เป็นเครื่องมือสำหรับการดาวน์โหลดโปรแกรมถือได้ว่าเป็นวิธีการที่สะดวกและประหยัดค่าใช้จ่ายในขั้นตอนการพัฒนาโปรแกรมประยุกต์ ผู้พัฒนาไม่จำเป็นต้องซื้อเครื่องโปรแกรมเพิ่ม เฟิร์มแวร์ดังกล่าวมีลักษณะเป็นโปรแกรมขนาดเล็ก ขนาดประมาณ 1-2 กิโลไบต์ มีชื่อเรียกว่า Bootloader ซึ่งผู้ผลิตฮาร์ดแวร์อาจพัฒนาเตรียมไว้ให้ใช้งานโดยติดตั้งไว้บนหน่วยความจำล่วงหน้าก่อนจำหน่ายฮาร์ดแวร์หรือผู้ใช้สามารถพัฒนาขึ้นมาใช้งานเองได้โดยใช้ภาษาโปรแกรมภาษาซี/ซี++ แล้วทำการติดตั้งลงบนหน่วยความจำด้วยเครื่องโปรแกรมในหัวข้อ 4.2.1 เฟิร์มแวร์ดังกล่าวจะช่วยดาวน์โหลดโปรแกรมที่พัฒนาเสร็จแล้วจากคอมพิวเตอร์ผ่านพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ การทำงานจะเริ่มเมื่อไมโครคอนโทรลเลอร์ได้รับสัญญาณรีเซต (reset signal) ทั้งจากภายในและภายนอกวงจรรวม จากนั้นจะรอการสื่อสารข้อมูลจากคอมพิวเตอร์เพื่อรอรับข้อมูลโปรแกรมใหม่ การใช้งาน Bootloader มีจุดเด่นในเรื่องการดาวน์โหลดโปรแกรมที่สามารถ

ทำได้โดยใช้คอมพิวเตอร์ทั่วไปและอาจประยุกต์ใช้กับการดาวน์โหลดโปรแกรมผ่านเครื่องรับส่งสัญญาณวิทยุ (over the air programming – OTA) ได้อีกด้วย แต่การใช้งานจะต้องเสียพื้นที่ในหน่วยความจำไปบางส่วนเพื่อเก็บโปรแกรม Bootloader และระยะเวลาในช่วงเริ่มทำงานที่จะเพิ่มขึ้น เนื่องจากโปรแกรม Bootloader รอการสื่อสารกับคอมพิวเตอร์เพื่อดาวน์โหลดโปรแกรมก่อนเริ่มการทำงานในส่วนโปรแกรมประยุกต์ผู้ใช้ ดังแสดงในแผนผังการทำงานในรูปที่ 4.2



รูปที่ 4.2 แผนผังการทำงานของโปรแกรม Bootloader

ต้นแบบโนดเซนเซอร์ในงานวิจัยนี้ออกแบบให้สามารถเชื่อมต่อกับคอมพิวเตอร์ได้ทั้ง 2 ช่องทาง ได้แก่ การเชื่อมต่อโดยใช้เครื่องโปรแกรม และ การเชื่อมต่อผ่านพอร์ตอนุกรม ทำให้สามารถเลือกวิธีดาวน์โหลดโปรแกรมลงบนโนดเซนเซอร์ได้ทั้ง 2 วิธีที่กล่าวมาข้างต้น ผู้ใช้สามารถเลือกใช้โดยพิจารณาจุดเด่นของแต่ละวิธีไปใช้งานได้ตามความต้องการใช้งาน การโปรแกรมโดยเครื่องโปรแกรมเลือกใช้เครื่องโปรแกรมของบริษัท Atmel มีชื่อว่า AVR ISP mkII แสดงในรูปที่ 4.3 เชื่อมต่อกับคอมพิวเตอร์ผ่านพอร์ตยูเอสบี สามารถดาวน์โหลดโปรแกรมลงไมโครคอนโทรเลอร์ตระกูล AVR โดยผ่านขาสัญญาณของบัสเอสพีไอ สามารถกำหนดค่าเรจิสเตอร์พิเศษเพื่อกำหนดการทำงานได้โดยตรง เครื่องโปรแกรมดังกล่าวไม่ต้องถอดไมโครคอนโทรลเลอร์มาติดตั้งบนเครื่องโปรแกรมโดยตรง พื้นที่หน่วยความจำทั้งหมดสามารถนำไปใช้เก็บโปรแกรมประยุกต์ได้ ในส่วนการโปรแกรมโดยใช้เฟิร์ม

แวร์นั้นต้นแบบโนดเซนเซอร์ถูกออกแบบให้สามารถติดตั้งเฟิร์มแวร์ Bootloader สำหรับช่วยดาวน์โหลดโปรแกรมผ่านพอร์ตอนุกรมได้เช่นเดียวกับบอร์ดอาดูอิโนมาตรฐาน และ Waspmote สามารถใช้งานกับคอมพิวเตอร์ทั่วไปได้และปรับแต่ง Bootloader ได้เอง ซึ่งมีความยืดหยุ่นมากกว่าการใช้งาน Waspmote ที่ไม่อนุญาตให้ปรับแต่งและไม่สามารถใช้เครื่องโปรแกรมสำหรับโปรแกรมได้อีกด้วย

รูปที่ 4.3 เครื่องโปรแกรม AVR ISP mkII

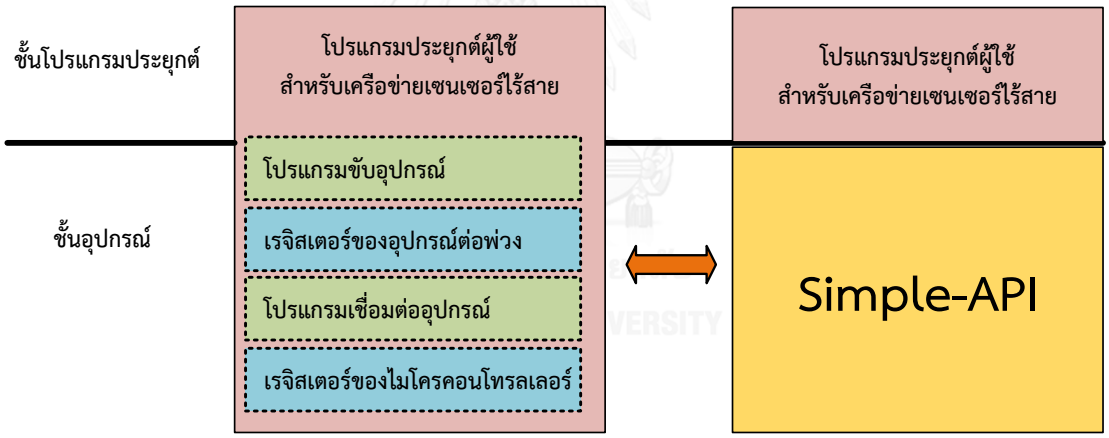
4.3 แนวคิดการพัฒนาส่วนต่อประสานโปรแกรมประยุกต์

การพัฒนาโปรแกรมประยุกต์สำหรับใช้งานโนดเซนเซอร์ที่กล่าวมาในหัวข้อ 4.1 พบว่าผู้ใช้งานโนดเซนเซอร์อาจต้องพัฒนาโปรแกรมในทุกระดับขึ้นมาใช้งานเองตั้งแต่โปรแกรมในระดับล่างที่ต้องจัดการการเชื่อมต่อและเรจิสเตอร์จนถึงโปรแกรมประยุกต์ของผู้ใช้ ผู้พัฒนาที่ไม่มีคุ้นเคยกับรูปแบบการพัฒนาอาจเขียนคำสั่งในส่วนโปรแกรมประยุกต์และโปรแกรมขับอุปกรณ์รวมกันเป็นส่วนเดียวซึ่งจะเป็นข้อเสียในการปรับปรุงหรือพัฒนาต่อในอนาคตและหากต้องการเปลี่ยนอุปกรณ์ชิ้นใหม่จะต้องทำการเปลี่ยนคำสั่งในโปรแกรมที่พัฒนาเสร็จแล้วทั้งหมดซึ่งทำได้ยากและเสียเวลา

การนำแพลตฟอร์มอาดูอิโนมาประยุกต์ใช้งานจะช่วยเรื่องการเขียนโปรแกรมการเชื่อมต่อและโปรแกรมขับอุปกรณ์บางตัวที่เตรียมไว้ให้ใช้ล่วงหน้า ไม่ต้องเข้าไปกำหนดค่าในระดับเรจิสเตอร์เองทั้งหมด สร้างขึ้นแยกเป็นคลาสตามประเภทการเชื่อมต่อและอุปกรณ์ตัวนั้น แต่หากผู้ใช้งานจะต้องจัดการการเชื่อมต่อและเรียกใช้อุปกรณ์เอง รวมถึงป้องกันการเรียกใช้งานและความผิดพลาดที่อาจจะเกิดขึ้นได้ ผู้ใช้งานที่ไม่คุ้นเคยกับลำดับการสั่งงานหรือตั้งค่าอาจทำให้เรียกใช้ฟังก์ชันต่าง ๆ ได้ไม่ครบอาจทำให้อุปกรณ์นั้นไม่สามารถใช้งานหรือทำงานไม่ถูกต้อง ในส่วนการใช้งาน Waspmote แม้ว่าทางผู้ผลิตจะเตรียมโปรแกรมขับอุปกรณ์ไว้ให้ใช้งานแล้ว แต่การใช้งานจริงผู้ใช้ยังคงพบเจอกับฟังก์ชันจำนวนมากในการใช้งานอุปกรณ์ ฟังก์ชันที่ถูกใช้งานเป็นการเรียกใช้อุปกรณ์ตัวนั้นโดยตรง ทำให้คำสั่งที่ใช้ในโปรแกรมประยุกต์ผู้ใช้ยึดติดกับอุปกรณ์ตัวนั้น การเปลี่ยนไปอุปกรณ์ตัวอื่นทำได้ยาก

เนื่องจากต้องเสียเวลาแก้ไขโปรแกรมประยุกต์ใหม่ทั้งหมด เพื่อให้ผู้ใช้งานสามารถเรียนรู้การเรียกใช้งานได้สะดวก ลดระยะเวลาในขั้นตอนการพัฒนาให้สั้นลงและสามารถพัฒนาปรับปรุงประสิทธิภาพการทำงานของโนดเซนเซอร์ งานวิจัยนี้จึงนำเสนอส่วนต่อประสานโปรแกรมประยุกต์ (application programming interface – API) เรียกชื่อว่า “Simple-API” ทำหน้าที่เป็นตัวเชื่อมระหว่างโปรแกรมประยุกต์ผู้ใช้และการติดต่อใช้งานอุปกรณ์บนโนดเซนเซอร์ดังแสดงในรูปที่ 4.4 ซึ่งมีแนวทางนำเสนอดังต่อไปนี้

- สร้างและจัดกลุ่มฟังก์ชันการทำงานโดยซ่อนการเขียนโปรแกรมระดับล่างและการเข้าถึงเรจิสเตอร์ของอุปกรณ์ที่ผู้ใช้ไม่จำเป็นต้องปรับเปลี่ยนในการใช้งาน
- ลดการเรียกใช้ฟังก์ชันที่มีชื่อต่างกันจำนวนมาก การเรียกใช้ชุดคำสั่งที่ไม่ผูกแน่นติดกับโปรแกรมประยุกต์ สามารถปรับเปลี่ยนการทำงานภายในฟังก์ชันได้โดยไม่ส่งผลกระทบต่อการเรียกใช้ภายโปรแกรมประยุกต์
- ป้องกันการความผิดพลาดในการเรียกใช้งานและระหว่างอุปกรณ์ทำงานตลอดจนแจ้งเตือนการทำงานที่ผิดพลาดด้วยรหัสตัวเลขแทนความผิดพลาด

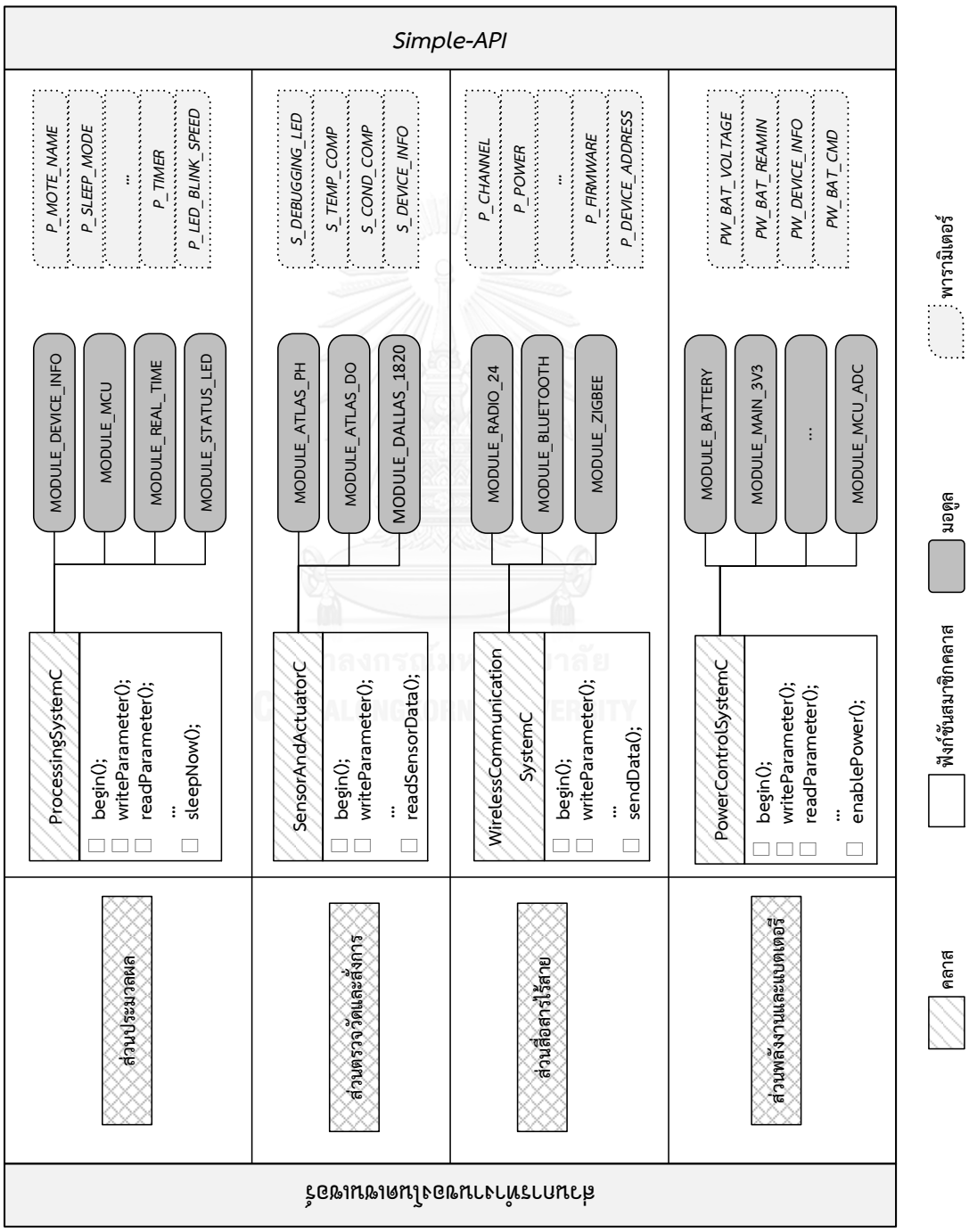


รูปที่ 4.4 ลำดับชั้นของการทำงานของ Simple-API

4.4 การออกแบบ Simple-API

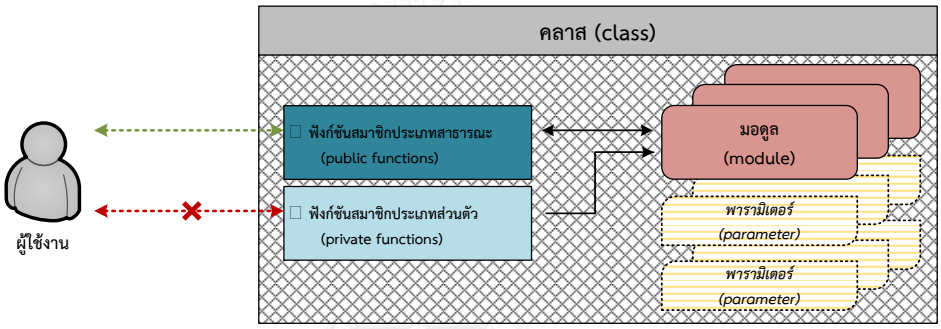
ส่วนต่อประสานโปรแกรมประยุกต์ Simple-API ที่นำเสนอในงานวิจัยชิ้นนี้ถูกออกแบบและพัฒนาขึ้นโดยอ้างอิงกับโครงสร้างการทำงานของโนดเซนเซอร์ที่กล่าวมาในหัวข้อที่ 2.3 ทั้ง 4 ส่วน (ส่วนประมวลผล ส่วนตรวจวัดและสั่งการ ส่วนสื่อสารไร้สาย และ ส่วนควบคุมพลังงานและแบตเตอรี่) โดยใช้หลักการเขียนโปรแกรมเชิงวัตถุ (object oriented programming - OOP) ร่วมกับรูปแบบการสร้างคลังโปรแกรมในแพลตฟอร์มอาดูอิโนด้วยภาษาโปรแกรมภาษาซี++

ส่วนการทำงานทั้งหมดจะถูกนำมาสร้างเป็นคลาส ได้แก่ คลาส ProcessingSystemC, คลาส SensorAndActuatorC, คลาส WirelessCommunicationC, และ PowerControlSystemC ภายในแต่ละคลาสนำเสนอแนวคิดการสร้างสัญลักษณ์เรียกว่า มอดูล (module) แทนอุปกรณ์ที่นำมาใช้งานในส่วนการทำงาน ร่วมกับสัญลักษณ์เรียกว่า พารามิเตอร์ (parameter) แทนค่าที่ใช้สำหรับปรับแต่งการทำงานหรือค่าที่อ่านได้ภายในอุปกรณ์นั้นดังแสดงในรูปที่ 4.5



รูปที่ 4.5 ภาพรวมโครงสร้างการออกแบบพัฒนา Simple-API

การออกแบบฟังก์ชันสมาชิกในแต่ละคลาสจะนำคุณสมบัติการห่อหุ้ม (encapsulation) ของการเขียนโปรแกรมเชิงวัตถุโดยฟังก์ชันสมาชิกประเภทส่วนตัว (private function) จะใช้เก็บค่าเริ่มต้นสำหรับการทำงานและชุดคำสั่งการเชื่อมต่ออุปกรณ์ที่ผู้ใช้ไม่จำเป็นต้องเข้าไปแก้ไข ในส่วนการควบคุมและสั่งงานจะทำผ่านฟังก์ชันสมาชิกประเภทสาธารณะ (public function) ซึ่งสามารถจัดกลุ่มตามทำหน้าที่ ได้แก่ การสั่งเริ่มการทำงาน การตั้งค่าหรืออ่านค่า และ การสั่งงานอุปกรณ์ เป็นต้น นอกจากนี้การพัฒนายังนำคุณสมบัติโอเวอร์โหลดฟังก์ชัน (function overloading) ทำให้สามารถออกแบบชื่อฟังก์ชันที่เหมือนกันได้แม้ว่าพารามิเตอร์ของฟังก์ชันจะแตกต่างกันก็ตาม ผู้ใช้งานจะมีความสะดวกในการเรียกใช้โดยไม่ต้องเรียนรู้และจำชื่อฟังก์ชันจำนวนมากที่มีหน้าที่การทำงานเหมือนกันแต่มีพารามิเตอร์ตางชนิดกัน

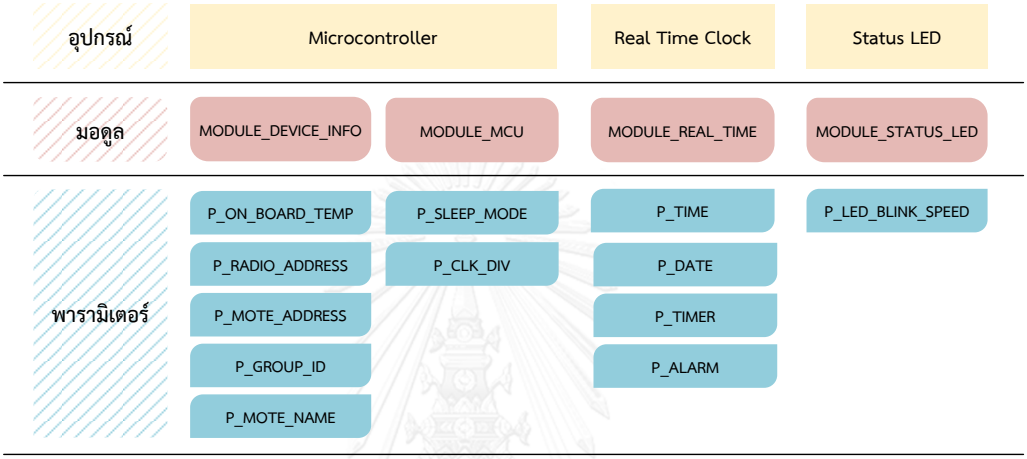


รูปที่ 4.6 การเข้าถึงมอดูลและพารามิเตอร์ภายในคลาส

การเรียกใช้ส่วนต่อประสานโปรแกรมประยุกต์ Simple-API มีรูปแบบการเรียกใช้เหมือนการใช้งานคลังโปรแกรมของแพลตฟอร์มอาดูอิโน ซึ่งจะสร้างวัตถุจากคลาส (รูปแบบคล้ายกับการประกาศตัวแปรในโปรแกรมทั่วไป) ขึ้นมาใช้ภายในโปรแกรม การใช้งานอุปกรณ์จะเรียกผ่านฟังก์ชันสมาชิกประเภทสาธารณะของวัตถุ ข้อแตกต่างระหว่าง Simple-API และโปรแกรมขับอุปกรณ์ของทั้งอาดูอิโนและ Waspmote คือการตรวจสอบความผิดพลาดจากการเรียกใช้งานโดยตรวจสอบการเรียกใช้ผ่าน ชื่อมอดูล ชื่อพารามิเตอร์ และ ค่าของพารามิเตอร์ ว่าการเรียกใช้อนุญาตให้ใช้งานหรือไม่ หรือค่าของพารามิเตอร์อยู่ในช่วงที่ถูกต้องหรือไม่ หากสามารถใช้งานได้ก็จะเข้าไปทำงานในชุดคำสั่งที่เขียนไว้ แต่หากไม่สามารถใช้งานได้หรือเกิดข้อผิดพลาดขึ้น ฟังก์ชันจะการคืนค่าผลของการทำงานในรูปแบบตัวเลขกลับมา เพื่อให้ผู้ใช้สามารถตรวจสอบการเรียกใช้งานฟังก์ชันนั้นได้และป้องกันโนดเซนเซอร์ทำงานค้างเนื่องจากโปรแกรมทำงานติดอยู่ภายในการวนรอบแบบอนันต์ด้วย รายละเอียดการออกแบบส่วนประกอบภายในและฟังก์ชันของทั้ง 4 คลาสของส่วนต่อประสานโปรแกรมประยุกต์ Simple-API มีดังต่อไปนี้

4.4.1 คลาส ProcessingSystemC

คลาส ProcessingSystemC ออกแบบให้รับผิดชอบการทำงานในส่วนประมวลผล ทำหน้าที่สั่งการควบคุมการทำงานอุปกรณ์ที่ใช้งานในส่วนประมวลผล ได้แก่ ไมโครคอนโทรลเลอร์ วงจรฐานเวลาจริง และ หลอดแอลอีดี คลาสดังกล่าวนี้จัดเป็นคลาสที่มีความสำคัญลำดับแรกของการสั่งงานอุปกรณ์บนโนดเซนเซอร์ การออกแบบมอดูลและพารามิเตอร์ที่ใช้งานภายในคลาสโครงสร้างแสดงในรูปที่ 4.7



รูปที่ 4.7 โครงสร้างการออกแบบภายในคลาส ProcessingSystemC

มอดูล MODULE_DEVICE_INFO ทำหน้าที่เก็บข้อมูลรายละเอียดของโนดเซนเซอร์ ข้อมูลในส่วนนี้จะถูกเก็บในหน่วยความจำ EEPROM ของไมโครคอนโทรลเลอร์ สามารถเก็บข้อมูลได้โดยไม่ต้องใช้พลังงานไฟฟ้า ภายในออกแบบให้มีพารามิเตอร์สำหรับใช้งานทั้งหมด 5 ตัวดังแสดงในตารางที่ 4.1

ตารางที่ 4.1 รายชื่อพารามิเตอร์ภายในมอดูล MODULE_DEVICE_INFO

พารามิเตอร์	รายละเอียด
P_MOTE_NAME	ชื่อของโนดเซนเซอร์ ข้อมูลจะเก็บอยู่ในรหัสแอสกี
P_GROUP_ID	หมายเลขประจำกลุ่มของโนดเซนเซอร์
P_MOTE_ADDRESS	หมายเลขประจำตัวโนดเซนเซอร์
P_RADIO_ADDRESS	หมายเลขประจำตัวของเครื่องรับส่งสัญญาณวิทยุที่ติดตั้งบนโนดเซนเซอร์
P_ON_BOARD_TEMP	อุณหภูมิบนแผงวงจรที่ตรวจวัดได้จากเซนเซอร์ภายในวงจรฐานเวลาจริง

มอดูล MODULE_MCU ทำหน้าที่กำหนดรูปแบบทำงานของไมโครคอนโทรลเลอร์ขณะใช้พลังงานต่ำและตัวหารความถี่ของสัญญาณนาฬิกาภายในไมโครคอนโทรลเลอร์ ภายในออกแบบให้มีพารามิเตอร์สำหรับใช้งานทั้งหมด 2 ตัวดังแสดงในตารางที่ 4.2

ตารางที่ 4.2 รายชื่อพารามิเตอร์ภายในมอดูล MODULE_MCU

พารามิเตอร์	รายละเอียด
P_DIV_CLK	ตัวหารความถี่ภายในของไมโครคอนโทรลเลอร์
P_SLEEP_MODE	โหมดการทำงานพลังงานต่ำของไมโครคอนโทรลเลอร์

มอดูล MODULE_REAL_TIME ทำหน้าที่เสมือนวงจรฐานเวลาจริงในโปรแกรม ให้ข้อมูลเวลาวันที่และใช้งานเป็นนาฬิการสำหรับตั้งปลุก การตั้งปลุกจะสามารถตั้งปลุกล่วงหน้าได้ 24 ชั่วโมง ในรูปแบบเวลาที่ต้องการและการตั้งจับเวลานับถอยหลัง ภายในออกแบบให้มีพารามิเตอร์สำหรับใช้งานทั้งหมด 4 ตัวดังแสดงในตารางที่ 4.3

ตารางที่ 4.3 รายชื่อพารามิเตอร์ภายในมอดูล MODULE_REAL_TIME

พารามิเตอร์	รายละเอียด
P_TIME	เวลาของวงจรฐานเวลาจริง
P_DATE	วันที่ของวงจรฐานเวลาจริง
P_ALARM	เวลาที่ใช้ตั้งปลุก (การทำงานเหมือนการตั้งนาฬิกาปลุกทั่วไป)
P_TIMER	ระยะเวลาตั้งปลุก (การทำงานเหมือนการตั้งนาฬิกาจับเวลาถอยหลัง)

มอดูล MODULE_STATUS_LED ทำหน้าที่กำหนดการกระพริบของหลอดแอลอีดีแสดงสถานะ ภายในออกแบบให้มีพารามิเตอร์สำหรับใช้งานทั้งหมด 1 ตัวดังแสดงในตารางที่ 4.4

ตารางที่ 4.4 รายชื่อพารามิเตอร์ภายในมอดูล MODULE_STATUS_LED

พารามิเตอร์	รายละเอียด
P_LED_BLINK_SPEED	เวลาของวงจรฐานเวลาจริง

การเข้าถึงอุปกรณ์สำหรับการเริ่มต้นการใช้งาน การปรับแต่งค่าการใช้งาน และการสั่งงานจะทำผ่านมอดูลและพารามิเตอร์ โดยใช้ฟังก์ชันสมาชิกของคลาสประเภทสาธารณะของคลาสเป็นผู้จัดการทั้งหมด การแบ่งกลุ่มและหน้าที่การทำงานของฟังก์ชันแสดงในตารางที่ 4.5 เมื่อเรียกใช้ฟังก์ชันเพื่อเข้าถึงอุปกรณ์จะมีการคืนค่ากลับมาเพื่อแสดงผลลัพธ์ของการทำงานสามารถตรวจสอบได้จากหมายเลขที่ใช้แทนความผิดพลาดที่เกิดขึ้นของการทำงาน

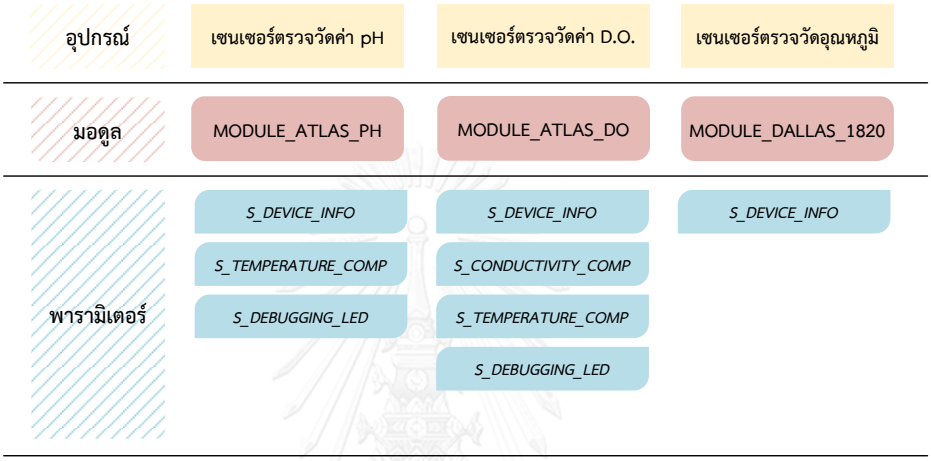
ตารางที่ 4.5 ฟังก์ชันสมาชิกแบบสาธารณะของคลาส ProcessingSystemC

ฟังก์ชันเริ่มการทำงาน	● ฟังก์ชันเริ่มการทำงานการเชื่อมต่อและกำหนดค่าเริ่มต้นของอุปกรณ์ สร้างการเชื่อมต่อไปยังวงจรฐานเวลาจริง การตั้งค่าการรับสัญญาณอินเทอร์รัพท์สำหรับการปลุกไมโครคอนโทรลเลอร์จากโหมดนอนหลับและกำหนดการทำงานของหลอดแอลอีดีแสดงสถานะบนแผงวงจร
ฟังก์ชันจัดการพารามิเตอร์	● ฟังก์ชันกำหนดค่าพารามิเตอร์ของมอดูล สามารถเรียกใช้งานได้ตลอดทั้งโปรแกรมเมื่อต้องการเปลี่ยนแปลงค่าพารามิเตอร์ของมอดูลนั้น
	● ฟังก์ชันอ่านค่าพารามิเตอร์ของมอดูล สามารถเรียกใช้งานได้ตลอดทั้งโปรแกรมเมื่อต้องการอ่านค่าพารามิเตอร์ของมอดูลนั้น
ฟังก์ชันสั่งงานอุปกรณ์	● ฟังก์ชันการเขียนข้อมูลผู้ใช้ลงในหน่วยความจำประเภท EEPROM โดยเขียนข้อมูลลงบล็อกในหน่วยความจำทีละ 10 ไบต์ จำนวน 350 ช่อง
	● ฟังก์ชันการอ่านข้อมูลผู้ใช้จากในหน่วยความจำประเภท EEPROM โดยอ่านข้อมูลจากบล็อกในหน่วยความจำทีละ 10 ไบต์ จำนวน 350 ช่อง
	● ฟังก์ชันเปิดใช้งานหลอดแอลอีดีแสดงสถานะบนแผงวงจรหลัก
	● ฟังก์ชันปิดใช้งานหลอดแอลอีดีแสดงสถานะบนแผงวงจรหลัก
	● ฟังก์ชันควบคุมจำนวนครั้งในการกระพริบหลอดแอลอีดีแสดงสถานะบนแผงวงจรหลัก
	● ฟังก์ชันสั่งไมโครคอนโทรเลอร์เข้าสู่การทำงานในโหมดนอนหลับ
	● ฟังก์ชันรีเซตการทำงานของไมโครคอนโทรลเลอร์

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

4.4.2 คลาส SensorAndActuatorC

คลาส SensorAndActuatorC ออกแบบให้รับผิดชอบการทำงานในส่วนตรวจวัดและสั่งการ ทำหน้าที่เชื่อมต่อกับเซนเซอร์และอุปกรณ์ที่ใช้สั่งงาน เพื่อให้สอดคล้องกับตัวอย่างเซนเซอร์ที่นำมาใช้เป็นตัวอย่างสำหรับงานตรวจวัดคุณภาพน้ำ ภายในประกอบด้วยอุปกรณ์ที่ใช้งานได้แก่ เซนเซอร์ตรวจวัดค่า pH เซนเซอร์ตรวจวัดค่าการละลายออกซิเจนในน้ำและเซนเซอร์ตรวจวัดอุณหภูมิ การออกแบบมอดูลและพารามิเตอร์ที่ใช้งานภายในคลาสโครงสร้างแสดงในรูปที่ 4.8



รูปที่ 4.8 โครงสร้างการออกแบบภายในคลาส SensorAndActuatorC

มอดูล MODULE_ATLAS_PH ทำหน้าที่เปรียบเสมือนชุดอุปกรณ์เซนเซอร์ตรวจวัดค่าพีเอชของบริษัท AtlasScientific สำหรับตรวจวัดความเป็นกรด-ด่างของสายละลาย ภายในออกแบบให้มีพารามิเตอร์สำหรับใช้งานทั้งหมด 3 ตัวดังแสดงในตารางที่ 4.6

ตารางที่ 4.6 รายชื่อพารามิเตอร์ภายในมอดูล MODULE_ATLAS_PH

พารามิเตอร์	รายละเอียด
S_DEBUGGING_LED	เปิดหรือปิดการใช้งานหลอดแอลอีดีเพื่อตรวจสอบการทำงานบนวงจรเซนเซอร์
S_TEMPERATURE_COMP	ค่าอุณหภูมิของสารละลายที่ทำการวัดเพื่อใช้เป็นข้อมูลในการคำนวณหาค่าพีเอช ไม่จำเป็นต้องกำหนดค่าอุณหภูมิทุกครั้งที่ทำการวัดหากไม่ได้สั่งปิดการจ่ายพลังงานไฟฟ้าให้กับเซนเซอร์
S_DEVICE_INFO	ข้อมูลหมายเลขรุ่นและวันที่ของเฟิร์มแวร์ภายในวงจรเซนเซอร์

มอดูล MODULE_ATLAS_DO ทำหน้าที่เปรียบเสมือนชุดอุปกรณ์เซนเซอร์ตรวจวัดค่า D.O. ของบริษัท AtlasScientific สำหรับตรวจวัดการละลายออกซิเจนในน้ำ ภายในออกแบบให้มีพารามิเตอร์สำหรับใช้งานทั้งหมด 4 ตัวดังแสดงในตารางที่ 4.7

ตารางที่ 4.7 รายชื่อพารามิเตอร์ภายในมอดูล MODULE_ATLAS_DO

พารามิเตอร์	รายละเอียด
S_DEBUGGING_LED	เปิดหรือปิดการใช้งานหลอดแอลอีดีเพื่อตรวจสอบการทำงานบนวงจรเซนเซอร์
S_TEMPERATURE_COMP	ค่าอุณหภูมิของสารละลายที่ทำการวัดเพื่อใช้เป็นข้อมูลในการคำนวณหาค่าพีเอช ไม่จำเป็นต้องกำหนดค่าอุณหภูมิทุกครั้งที่ทำการวัดหากไม่ได้สั่งปิดการจ่ายพลังงานไฟฟ้าให้กับเซนเซอร์
S_CONDUCTIVITY_COMP	กำหนดค่าการนำไฟฟ้า (conductivity) ของน้ำที่ทำการวัดเพื่อเป็นข้อมูลในการคำนวณหาค่า D.O.
S_DEVICE_INFO	ข้อมูลหมายเลขรุ่นและวันที่ของเฟิร์มแวร์ภายในวงจรเซนเซอร์

มอดูล MODULE_DALLAS_1820 ทำหน้าที่เปรียบเสมือนตัวอุปกรณ์เซนเซอร์ตรวจวัดอุณหภูมิของบริษัท Dallas สำหรับตรวจวัดอุณหภูมิของน้ำในหน่วยองศาเซลเซียส ภายในออกแบบให้มีพารามิเตอร์สำหรับใช้งานทั้งหมด 1 ตัวดังแสดงในตารางที่ 4.8

ตารางที่ 4.8 รายชื่อพารามิเตอร์ภายในมอดูล MODULE_DALLAS_1820

พารามิเตอร์	รายละเอียด
S_DEVICE_INFO	เก็บข้อมูลหมายเลขเซนเซอร์

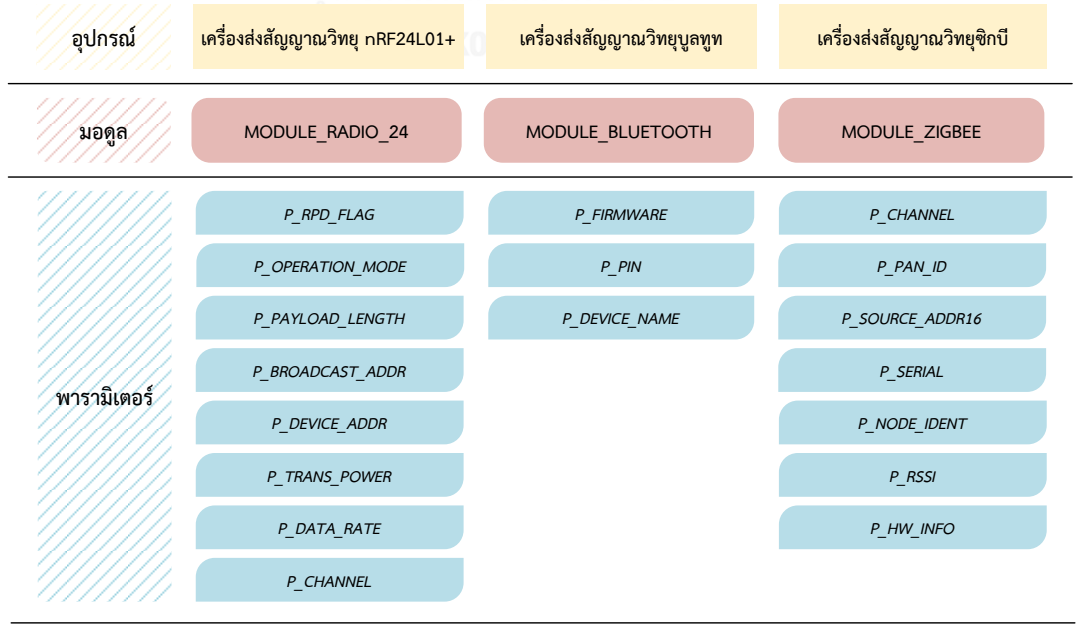
การเข้าถึงอุปกรณ์สำหรับการเริ่มต้นการใช้งาน การปรับแต่งค่าการใช้งาน และ การสั่งงานจะทำผ่านมอดูลและพารามิเตอร์ โดยใช้ฟังก์ชันสมาชิกของคลาสประเภทสาธารณะของคลาสเป็นผู้จัดการทั้งหมด การออกแบบชื่อและหน้าที่การทำงานของฟังก์ชันแสดงในตารางที่ 4.9 ฟังก์ชันสมาชิกแบบสาธารณะของคลาส SensorAndActuatorC เมื่อเรียกใช้ฟังก์ชันเพื่อเข้าถึงอุปกรณ์จะมีการคืนค่ากลับมาเพื่อแสดงผลลัพธ์ของการทำงานสามารถตรวจสอบได้จากหมายเลขที่ใช้แทนความผิดพลาดที่เกิดขึ้นของการทำงานเช่นเดียวกันกับคลาส ProcessingSystemC

ตารางที่ 4.9 ฟังก์ชันสมาชิกแบบสาธารณะของคลาส SensorAndActuatorC

ฟังก์ชันเริ่มการทำงาน	• ฟังก์ชันสำหรับเริ่มการเชื่อมต่อเซนเซอร์และกำหนดค่าเริ่มต้นการใช้งานอุปกรณ์ กลุ่มคำสั่งภายในจะสร้างการเชื่อมต่อไปยังอุปกรณ์จริงตามชื่อของมอดูลและช่องเชื่อมต่อ
ฟังก์ชันจัดการพารามิเตอร์	• ฟังก์ชันสำหรับกำหนดค่าพารามิเตอร์ของมอดูล สามารถเรียกใช้งานได้ตลอดทั้งโปรแกรม เมื่อต้องการเปลี่ยนแปลงค่าพารามิเตอร์ของมอดูลนั้น
	• ฟังก์ชันสำหรับอ่านค่าพารามิเตอร์ของมอดูล สามารถเรียกใช้งานได้ตลอดทั้งโปรแกรม เมื่อต้องการอ่านค่าพารามิเตอร์ของมอดูลนั้น
ฟังก์ชันสั่งงานอุปกรณ์	• ฟังก์ชันสำหรับอ่านค่าที่ได้จากการตรวจวัดของเซนเซอร์
	• ฟังก์ชันสำหรับปรับตั้งค่ามาตรฐานของเซนเซอร์

4.4.3 คลาส WirelessCommunicationSystemC

คลาส WirelessCommunicationSystemC ออกแบบให้รับผิดชอบการทำงานในส่วนการสื่อสารไร้สาย ทำหน้าที่เชื่อมต่อกับเครื่องรับส่งสัญญาณวิทยุที่ติดตั้งบนโนดเซนเซอร์ เพื่อให้สอดคล้องกับตัวอย่างเครื่องรับส่งสัญญาณวิทยุที่นำมาใช้งาน ได้แก่ เครื่องรับส่งสัญญาณวิทยุ เครื่องรับส่งสัญญาณวิทยุเทคโนโลยีบูลทูท เครื่องรับส่งสัญญาณวิทยุเทคโนโลยีซิกบี การออกแบบมอดูลและพารามิเตอร์ที่ใช้งานภายในคลาสโครงสร้างแสดงรูปที่ 4.9



รูปที่ 4.9 โครงสร้างการออกแบบภายในคลาส WirelessCommunicationSystemC

มอดูล MODULE_RADIO_24 ทำหน้าที่เปรียบเสมือนเครื่องรับส่งสัญญาณวิทยุ nRF24L01+ เครื่องรับส่งสัญญาณวิทยุจะส่งข้อมูลในรูปแบบเครือข่ายดาว (star network) ไปยังทุกเครื่องรับส่งสัญญาณวิทยุโดยสามารถกรองข้อมูลรับได้โดยใช้หมายเลขประจำตัวที่กำหนดไว้ ที่ภายในออกแบบให้มีพารามิเตอร์เพื่อใช้งานทั้งหมด 8 ตัว ดังแสดงตารางที่ 4.10

มอดูล MODULE_BLUETOOTH ทำหน้าที่เปรียบเสมือนเครื่องรับส่งสัญญาณวิทยุ HC-05 ใช้เทคโนโลยีบูลทูท ทำงานในโหมดสเลฟเท่านั้นการทำงานต้องรอการจับคู่อุปกรณ์มาสเตอร์เท่านั้น ภายในออกแบบให้มีพารามิเตอร์เพื่อใช้งานทั้งหมด 3 ตัว ดังแสดงในตารางที่ 4.11

ตารางที่ 4.10 รายชื่อพารามิเตอร์ภายในมอดูล MODULE_RADIO_24

พารามิเตอร์	รายละเอียด
P_CHANNEL	ช่องสัญญาณในการสื่อสารไร้สาย เครื่องส่งสัญญาณวิทยุที่จะต้องใช้ช่องสัญญาณเดียวกันในการสื่อสารระหว่างโนด
P_DATA_RATE	กำหนดความเร็วในการสื่อสารไร้สาย เครื่องส่งสัญญาณวิทยุที่จะต้องใช้ช่องสัญญาณเดียวกันในการสื่อสารระหว่างโนด
P_TRANS_POWER	กำลังที่ใช้ส่งสัญญาณวิทยุ เครื่องส่งสัญญาณวิทยุที่จะต้องใช้ช่องสัญญาณเดียวกันในการสื่อสารระหว่างโนด
P_DEVICE_ADDR	หมายเลขประจำเครื่องส่งสัญญาณวิทยุ ใช้สำหรับกรองข้อมูลที่รับมา
P_BROADCAST_ADDR	หมายเลขสำหรับกรองข้อความบรอดคาสต์ (broadcast) ภายในเครือข่าย
P_PAYLOAD_LENGTH	กำหนดความยาวของข้อมูลที่ใช้ส่ง เครื่องส่งสัญญาณวิทยุที่จะต้องมีการกำหนดความยาวของข้อมูลขนาดเดียวกันในการสื่อสารระหว่างโนด
P_OPERATION_MODE	กำหนดโหมดการทำงานของเครื่องส่งสัญญาณวิทยุ
P_RDP_FLAG	แสดงค่าพลังงานของช่องสัญญาณ

ตารางที่ 4.11 รายชื่อพารามิเตอร์ภายในมอดูล MODULE_BLUETOOTH

พารามิเตอร์	รายละเอียด
P_DEVICE_NAME	กำหนดชื่อเครื่องรับส่งสัญญาณวิทยุ โดยจะใช้เป็นชื่อในขั้นตอนการจับคู่อุปกรณ์ สามารถเขียนค่าได้เพียงอย่างเดียวและทำขณะเครื่องรับส่งสัญญาณยังไม่ได้จับคู่อุปกรณ์
P_PIN	กำหนดรหัสผ่าน 4 ตัวสำหรับการจับคู่อุปกรณ์ สามารถเขียนค่าได้เพียงอย่างเดียวและทำขณะเครื่องรับส่งสัญญาณยังไม่ได้จับคู่อุปกรณ์
P_FIRMWARE	เฟิร์มแวร์ภายในเครื่องรับส่งสัญญาณวิทยุ สามารถอ่านค่าได้เพียงอย่างเดียวและทำขณะเครื่องรับส่งสัญญาณยังไม่ได้จับคู่อุปกรณ์

มอดูล MODULE_ZIGBEE ทำหน้าที่เปรียบเสมือนเครื่องรับส่งสัญญาณวิทยุ Xbee PRO/Zigbee ใช้เทคโนโลยีซิกบีในการจัดการการสื่อสาร ภายในออกแบบให้มีพารามิเตอร์เพื่อใช้งานทั้งหมด 7 ตัว ดังแสดงในตารางที่ 4.12

ตารางที่ 4.12 รายชื่อพารามิเตอร์ภายในมอดูล MODULE_ZIGBEE

พารามิเตอร์	รายละเอียด
P_CHANNEL	ช่องสัญญาณในการสื่อสารไร้สาย เครื่องส่งสัญญาณวิทยุที่จะต้องใช้ช่องสัญญาณเดียวกันในการสื่อสารระหว่างโนด
P_PAN_ID	หมายเลขเครือข่ายส่วนตัว (personal area network)
P_SOURCE_ADDR	หมายเลขประจำตัวโนด (16 บิต)
P_SERIAL	หมายเลขประจำเครื่องส่งสัญญาณวิทยุ (64 บิต)
P_NODE_IDENT	ชื่อโนดเก็บในรูปแบบข้อความรหัสแอสกี
P_RSSI	ค่าระดับสัญญาณ RSSI ในการรับแพ็กเกต
P_HW_INFO	หมายเลขรุ่นอุปกรณ์

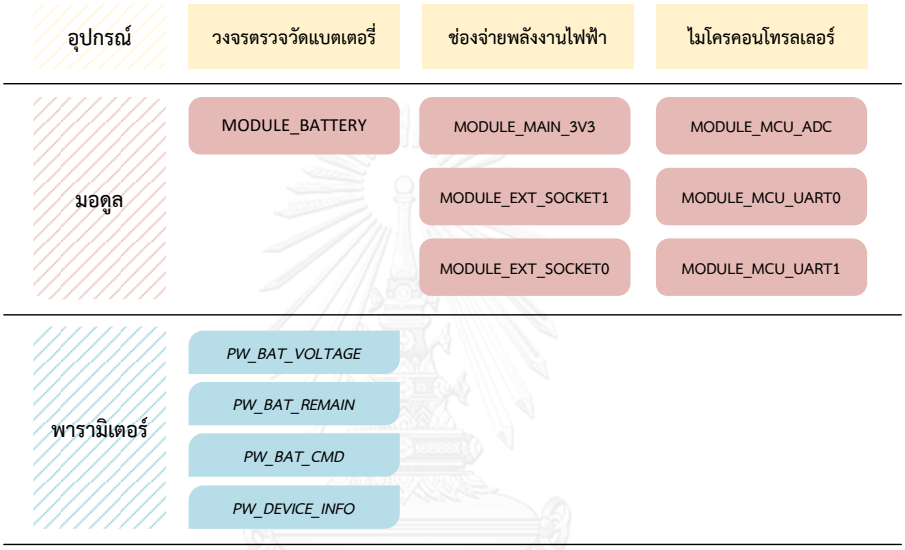
การเข้าถึงอุปกรณ์จริงสำหรับการเริ่มต้นการใช้งาน การปรับแต่งค่าการใช้งาน และการสั่งงานจะทำผ่านมอดูลและพารามิเตอร์ โดยใช้ฟังก์ชันสมาชิกของคลาสประเภทสาธารณะของคลาสเป็นผู้จัดการทั้งหมด การออกแบบชื่อและหน้าที่การทำงานของฟังก์ชันแสดงในตารางที่ 4.13 ผลการทำงานสามารถตรวจสอบได้โดยดูจากรหัสหมายเลขที่ใช้แทนความผิดพลาดของการทำงาน เมื่อฟังก์ชันทำงานเสร็จจะมีการคืนค่ากลับมาเพื่อแสดงผลลัพธ์ของการทำงานในคลาสนี้รหัสตัวเลขที่คืนค่ากลับหลังจากฟังก์ชันทำงานเสร็จจะแทนความหมายของการทำงาน

ตารางที่ 4.13 ฟังก์ชันสมาชิกแบบสาธารณะของคลาส WirelessCommunicationSystemC

ฟังก์ชันเริ่มการทำงาน	• ฟังก์ชันเริ่มการเชื่อมต่อและกำหนดค่าเริ่มต้นของอุปกรณ์ คำสั่งภายในจะสร้างการเชื่อมต่อไปยังอุปกรณ์จริงตามชื่อของมอดูลและช่องเชื่อมต่อ
ฟังก์ชันจัดการพารามิเตอร์	• ฟังก์ชันกำหนดค่าพารามิเตอร์ของมอดูล สามารถเรียกใช้งานได้ตลอดทั้งโปรแกรมเมื่อต้องการเปลี่ยนแปลงค่าพารามิเตอร์ของมอดูลนั้น
	• ฟังก์ชันอ่านค่าพารามิเตอร์ของมอดูล สามารถเรียกใช้งานได้ตลอดทั้งโปรแกรมเมื่อต้องการอ่านค่าพารามิเตอร์ของมอดูลนั้น
ฟังก์ชันสั่งงานอุปกรณ์	• ฟังก์ชันส่งข้อมูลที่ส่งผ่านทางเครื่องรับส่งสัญญาณวิทยุ สามารถเรียกใช้งานได้ตลอดทั้งโปรแกรมเมื่อต้องการอ่านค่าพารามิเตอร์ของมอดูลนั้น
	• ฟังก์ชันอ่านข้อมูลที่รับผ่านทางเครื่องรับส่งสัญญาณวิทยุ สามารถเรียกใช้งานได้ตลอดทั้งโปรแกรมเมื่อต้องการอ่านค่าพารามิเตอร์ของมอดูลนั้น

4.4.4 คลาส PowerControlSystemC

คลาส PowerControlSystemC ออกแบบให้รับผิดชอบการทำงานในส่วนการจัดการข้อมูลของแบตเตอรี่ผ่านวงจรรวมตรวจสอบแบตเตอรี่ การเปิดปิดการทำงานของวงจรพิเศษในไมโครคอนโทรลเลอร์ และ การจ่ายพลังงานไฟฟ้าบนแผงวงจรส่วนขยายสำหรับอุปกรณ์ต่อพ่วง ภายในคลาสประกอบด้วยสัญลักษณ์และกลุ่มฟังก์ชันที่ใช้สำหรับสั่งงานอุปกรณ์จริงดังนี้ การออกแบบมอดูลและพารามิเตอร์ที่ใช้งานภายในคลาสโครงสร้างแสดงในรูปที่ 4.10 โดยมีรายละเอียดของแต่ละมอดูลดังต่อไปนี้



รูปที่ 4.10 โครงสร้างการออกแบบภายในคลาส PowerControlSystemC

มอดูล MODULE_MCU_ADC ทำหน้าที่เปรียบเสมือนวงจรแปลสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัลภายในไมโครคอนโทรลเลอร์ มอดูลนี้ไม่มีพารามิเตอร์ปรับแต่งการทำงาน สามารถสั่งปิดการทำงานเพื่อลดการใช้พลังงานได้เมื่อไม่ต้องการใช้งาน

มอดูล MODULE_MCU_UART0 ทำหน้าที่เปรียบเสมือนวงจรการสื่อสารอนุกรมช่อง 0 ภายในไมโครคอนโทรลเลอร์ มอดูลนี้ไม่มีพารามิเตอร์ปรับแต่งการทางน สามารถสั่งปิดการทำงานเพื่อลดการใช้พลังงานได้เมื่อไม่ต้องการใช้งาน

มอดูล MODULE_MCU_UART1 ทำหน้าที่เปรียบเสมือนวงจรการสื่อสารอนุกรมช่อง 1 ภายในไมโครคอนโทรลเลอร์ มอดูลนี้ไม่มีพารามิเตอร์ปรับแต่งการทำงาน สามารถสั่งปิดการทำงานเพื่อลดการใช้พลังงานได้เมื่อไม่ต้องการใช้งาน

มอดูล MODULE_EXT_SOCKET0 ทำหน้าที่เปรียบเสมือนสวิตช์ควบคุมการเปิดปิดการจ่ายพลังงานไฟฟ้าให้กับช่องต่อที่ 0 บนแผงวงจรส่วนขยาย มอดูลนี้ไม่มีพารามิเตอร์ปรับแต่งการทำงาน สามารถสั่งปิดการทำงานเพื่อลดการใช้พลังงานได้เมื่อไม่ต้องการใช้งาน

มอดูล MODULE_EXT_SOCKET1 ทำหน้าที่เปรียบเสมือนสวิตช์ควบคุมการเปิดปิดการจ่ายพลังงานไฟฟ้าให้กับช่องต่อที่ 1 บนแผงวงจรส่วนขยาย มอดูลนี้ไม่มีพารามิเตอร์ปรับแต่งการทำงาน สามารถสั่งปิดการทำงานเพื่อลดการใช้พลังงานได้เมื่อไม่ต้องการใช้งาน

มอดูล MODULE_MAIN_3V3 ทำหน้าที่เปรียบเสมือนสวิตช์ของวงจรรวมรักษาแรงดันขนาด 3.3 โวลต์ บนแผงวงจรส่วนขยาย มอดูลนี้ไม่มีพารามิเตอร์ปรับแต่งการทำงาน สามารถสั่งปิดการทำงานเพื่อลดการใช้พลังงานได้เมื่อไม่ต้องการใช้งาน

มอดูล MODULE_BATTERY ทำหน้าที่เปรียบเสมือนวงจรตรวจสอบแบตเตอรี่ ให้ข้อมูลพลังงานของแบตเตอรี่ ภายในออกแบบให้มีพารามิเตอร์ทั้งหมด 4 ตัว ดังแสดงในตารางที่ 4.14

ตารางที่ 4.14 รายชื่อพารามิเตอร์ภายในมอดูล MODULE_BATTERY

พารามิเตอร์	รายละเอียด
PW_DEVICE_INFO	ข้อมูลหมายเลขรุ่นของวงจรรวมตรวจสอบแบตเตอรี่
PW_BAT_VOL	แรงดันไฟฟ้าของแบตเตอรี่ขาเข้าวงจรตรวจวัด (หน่วยโวลต์)
PW_BAT_REMAIN	ปริมาณประจุไฟฟ้าที่เหลืออยู่ในแบตเตอรี่ (หน่วยเปอร์เซ็นต์)
PW_BAT_CMD	คำสั่งพิเศษควบคุมการทำงานวงจรแบตเตอรี่

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

การเข้าถึงอุปกรณ์สำหรับการเริ่มต้นการใช้งาน การปรับแต่งค่าการใช้งาน และการสั่งงานจะทำผ่านมอดูลและพารามิเตอร์ โดยใช้ฟังก์ชันสมาชิกของคลาสประเภทสาธารณะของคลาสเป็นผู้จัดการทั้งหมด การออกแบบชื่อและหน้าที่การทำงานของฟังก์ชันแสดงในตารางที่ 4.15 เมื่อเรียกใช้ฟังก์ชันเพื่อเข้าถึงอุปกรณ์จะมีการคืนค่ากลับมาเพื่อแสดงผลลัพธ์ของการทำงานสามารถตรวจสอบได้จากหมายเลขที่ใช้แทนความผิดพลาดที่เกิดขึ้นของการทำงาน

ตารางที่ 4.15 ฟังก์ชันสมาชิกแบบสาธารณะของคลาส PowerControlSystemC

ฟังก์ชันเริ่มการทำงาน	• ฟังก์ชันสร้างการเชื่อมต่อไปยังวงจรตรวจวัดแบตเตอรี่และกำหนดค่าเริ่มต้นเปิดใช้งานวงจรพิเศษทุกวงจรในไมโครคอนโทรลเลอร์ ต้องเรียกใช้งานฟังก์ชันเป็นลำดับแรกเสมอก่อนการเรียกใช้ฟังก์ชันอื่น ๆ
ฟังก์ชันจัดการพารามิเตอร์	• ฟังก์ชันกำหนดค่าพารามิเตอร์ของมอดูล สามารถเรียกใช้งานได้ตลอดทั้งโปรแกรมเมื่อต้องการเปลี่ยนแปลงค่าพารามิเตอร์ของมอดูลนั้น
	• ฟังก์ชันอ่านค่าพารามิเตอร์ของมอดูล สามารถเรียกใช้งานได้ตลอดทั้งโปรแกรมเมื่อต้องการอ่านค่าพารามิเตอร์ของมอดูลนั้น
ฟังก์ชันสั่งงานอุปกรณ์	• ฟังก์ชันการเปิดใช้งานมอดูลภายในไมโครคอนโทรลเลอร์หรือการปิดใช้งานอุปกรณ์ภายนอกสามารถเรียกใช้งานได้ตลอดทั้ง
	• ฟังก์ชันการปิดใช้งานมอดูลภายในไมโครคอนโทรลเลอร์หรือการปิดใช้งานอุปกรณ์ภายนอกสามารถเรียกใช้งานได้ตลอดทั้ง

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

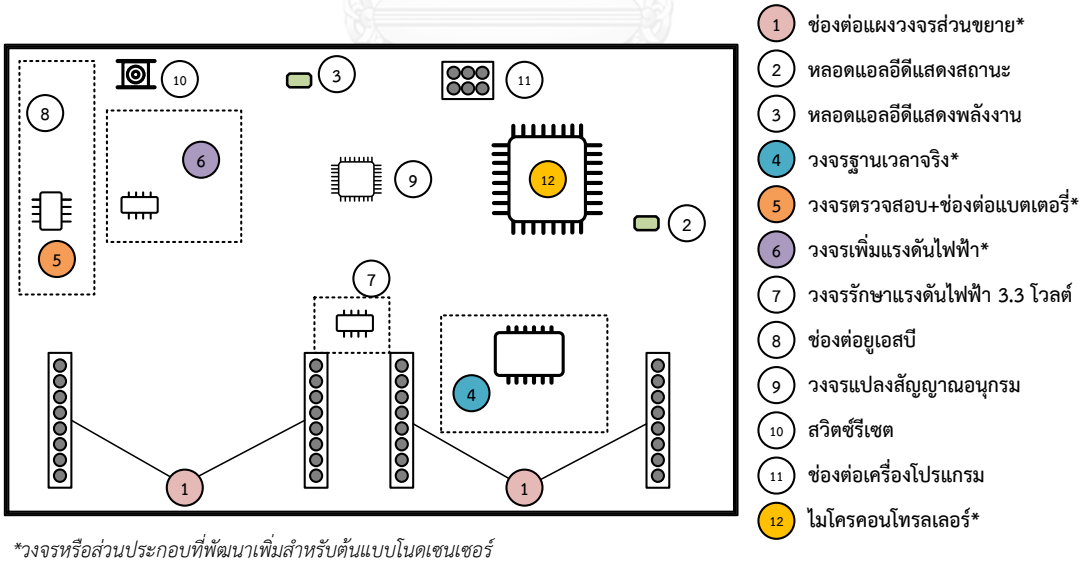
บทที่ 5
ผลการดำเนินงาน

5.1 ต้นแบบโนดเซนเซอร์

5.1.1 ลำดับการพัฒนาฮาร์ดแวร์ต้นแบบ

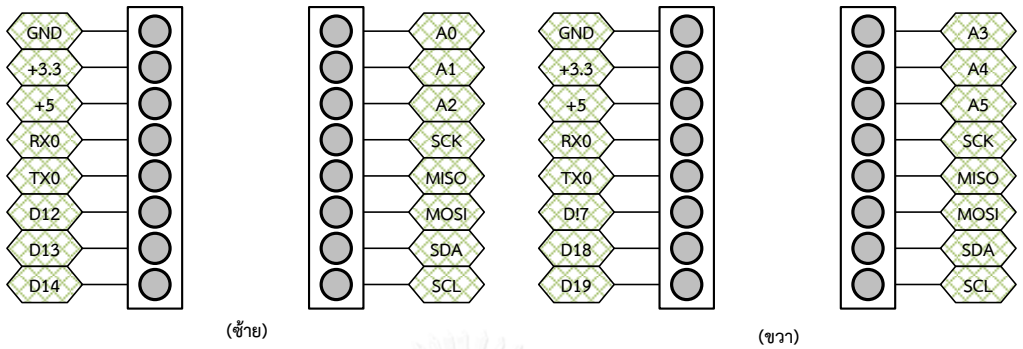
ต้นแบบโนดเซนเซอร์ที่พัฒนาขึ้นในงานวิจัยชิ้นนี้มีลำดับการพัฒนาโดยเริ่มจากการพัฒนาแผงวงจรหลักก่อนและพัฒนาแผงวงจรส่วนขยายขึ้นมาภายหลังโดยมีลำดับการพัฒนาดังต่อไปนี้

แผงวงจรหลักของต้นแบบโนดเซนเซอร์รุ่นที่ 1 มีขนาดกว้าง 85 มิลลิเมตร ยาว 150 มิลลิเมตร ออกแบบให้มีลายวงจรทั้งสองด้านของบอร์ด ไมโครคอนโทรลเลอร์และอุปกรณ์อิเล็กทรอนิกส์อื่น ๆ เลือกใช้อุปกรณ์ประเภทยึดติดบนแผงวงจรหรือเรียกว่าอุปกรณ์ประเภท SMD (surface mount device) มีโครงสร้างส่วนประกอบบนแผงวงจรหลักแสดงในรูปที่ 5.1 สามารถติดตั้งแผงวงจรส่วนต่อขยายได้ 2 แผง บนช่องต่อแผงวงจรส่วนต่อขยาย แผงวงจรหลักรุ่นที่ 1 มีจุดประสงค์ในการพัฒนาขึ้นเพื่อทดสอบการนำวงจรที่ต้องการนำมาใช้งานมาสร้างไว้บนแผงวงจรเดียวกัน ทดสอบการทำงานของไมโครคอนโทรเลอร์ การเชื่อมต่อเครื่องรับส่งสัญญาณวิทยุและเซนเซอร์ต่าง ๆ



รูปที่ 5.1 โครงสร้างภายในแผงวงจรหลักรุ่นที่ 1

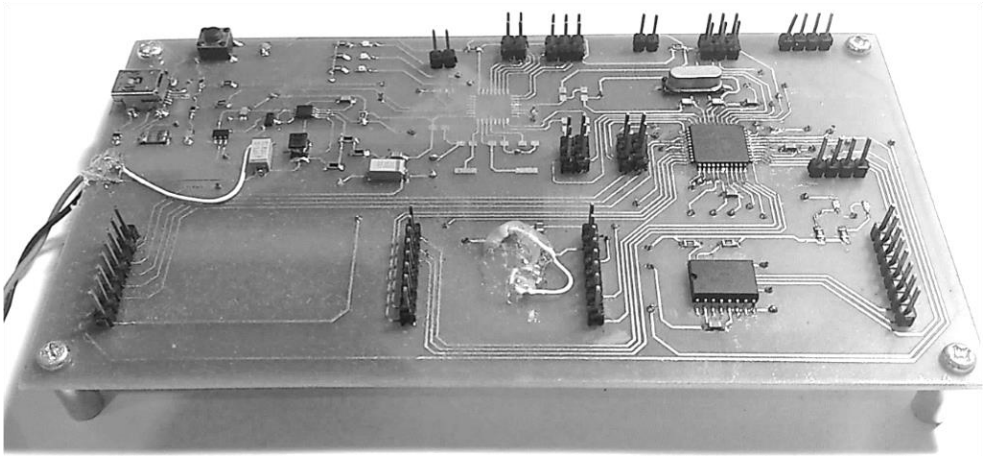
ช่องต่อแผงวงจรส่วนขยายทั้ง 2 ด้าน (ซ้ายและขวา) ออกแบบให้เป็นช่องต่อตัวผู้ เชื่อมต่อขาสัญญาณจากไมโครคอนโทรลเลอร์หลักออกมาโดยออกแบบให้มีช่องต่อดิจิทัลและช่องแอนะล็อก รวมถึงพอร์ตอนุกรม บัสเอสพีไอ และ บัส I^2C ดังแสดงในรูปที่ 5.2



รูปที่ 5.2 โครงสร้างช่องต่อแผงวงจรส่วนต่อขยายต้นแบบโนดเซนเซอร์รุ่นที่ 1

การทดสอบการทำงานต้นแบบโนดเซนเซอร์รุ่นที่ 1 พบว่าไมโครคอนโทรลเลอร์สามารถโปรแกรมด้วยเครื่องโปรแกรมได้ วงจรแปลงแรงดันไฟฟ้าสามารถเพิ่มแรงดันไฟฟ้าของแบตเตอรี่ขึ้นได้จนมีขนาด 5 โวลต์ แต่ยังมีข้อผิดพลาดในการทำงานได้แก่ ลายเส้นทองแดงบนแผงวงจรที่เชื่อมต่อวงจรเพิ่มระดับแรงดันไฟฟ้าและวงจรรักษาแรงดัน 3.3 โวลต์บางเส้นขาดหายไป วงจรรวมสำหรับแปลงรูปแบบการสื่อสารด้วยพอร์ตอนุกรมให้อยู่ในรูปแบบการสื่อสารด้วยช่องยูเอสบีไม่ทำงานรวมถึงผู้ผลิตมีการยกเลิกการผลิตวงจรรวมดังกล่าว ในระหว่างการนำมาทดสอบใช้งานผู้วิจัยยังพบปัญหาในการเปลี่ยนไมโครคอนโทรลเลอร์ตัวใหม่ได้ยากเมื่อเกิดความเสียหายขึ้น การเปลี่ยนไมโครคอนโทรลเลอร์อาจทำให้ลายวงจรเสียหายได้จากใช้เครื่องเป่าลมร้อนโดยไม่ระมัดระวัง

Chulalongkorn University

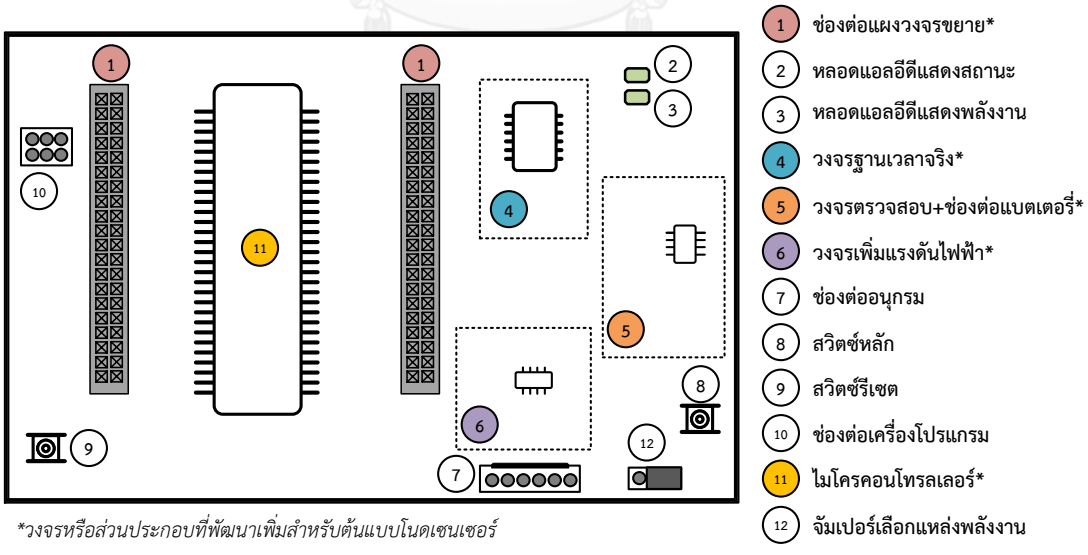


รูปที่ 5.3 แผงวงจรหลักรุ่นที่ 1

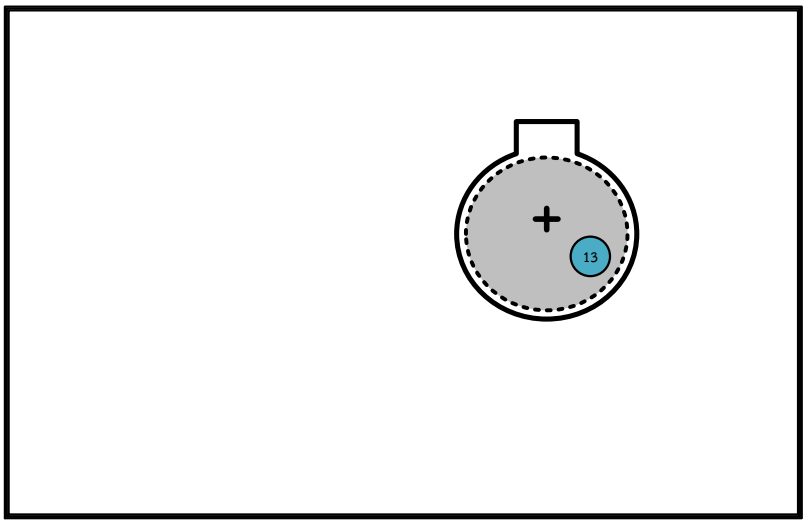
การแก้ไขข้อบกพร่องที่พบในการออกแบบและการทำงานที่ผิดพลาดของวงจรรวมในแผงวงจรหลักรุ่นที่ 1 ผู้วิจัยจึงเลือกแก้ไขโดยออกแบบแผงวงจรหลักใหม่ในรุ่นที่ 2 โดยทำการแก้ไขในประเด็นหลักดังต่อไปนี้

- การเปลี่ยนแพ็กเกจของไมโครคอนโทรลเลอร์จากเดิมประเภท (TQFP-44) เป็นประเภทขาเสียบ (DIP-40) ซึ่งสามารถถอดเปลี่ยนได้ง่ายเมื่อเกิดความเสียหายขึ้น และเพิ่มช่องสัญญาณของช่องเชื่อมต่อแผงวงจรส่วนขยายให้เข้าถึงขาสัญญาณของไมโครคอนโทรลเลอร์มากยิ่งขึ้น
- การเปลี่ยนวงจรรวมที่ใช้สำหรับแปลงรูปแบบการสื่อสารด้วยพอร์ตอนุกรมให้อยู่ในรูปแบบการสื่อสารด้วยช่องยูเอสบีเป็นวงจรรวม FTDI-232 โดยแยกเป็นวงจรภายนอกไม่ติดตั้งลงในแผงวงจรหลักเพื่อให้ลดต้นทุนในการสร้างบอร์ดหลักและสามารถเลือกใช้จากผู้ผลิตรายอื่นได้ อีกทั้งเมื่อนำโนดเซนเซอร์ไปใช้งานจริงก็ไม่มีความจำเป็นที่ต้องมีวงจรรวมดังกล่าวนี้
- การเพิ่มแบตเตอรี่สำรองสำหรับวงจรฐานเวลาจริงและการย้ายวงจรรวมสำหรับรักษาแรงดันขนาด 3.3 โวลต์ไปไว้บนส่วนต่อขยายจากของเดิมที่ติดตั้งไว้บนแผงวงจรหลักเพื่อให้สะดวกต่อการเปิดปิดการใช้งาน

แผงวงจรหลักในรุ่นที่ 2 มีโครงสร้างส่วนประกอบที่สำคัญแสดงในรูปที่ 5.4 และ รูปที่ 5.5



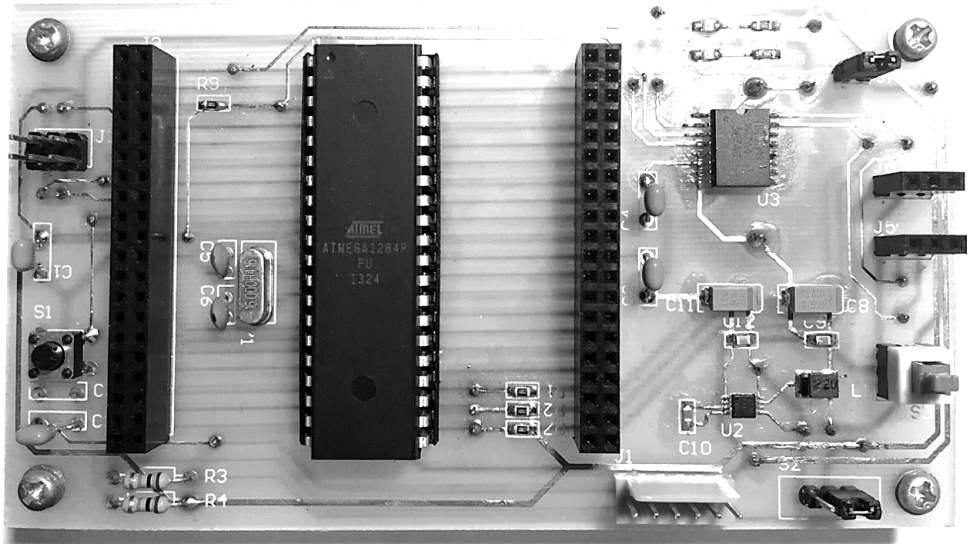
รูปที่ 5.4 โครงสร้างแผงวงจรของต้นแบบโนดเซนเซอร์หลักรุ่นที่ 2 ด้านหน้า



**วงจรหรือส่วนประกอบที่พัฒนาเพิ่มสำหรับต้นแบบโนดเซนเซอร์*

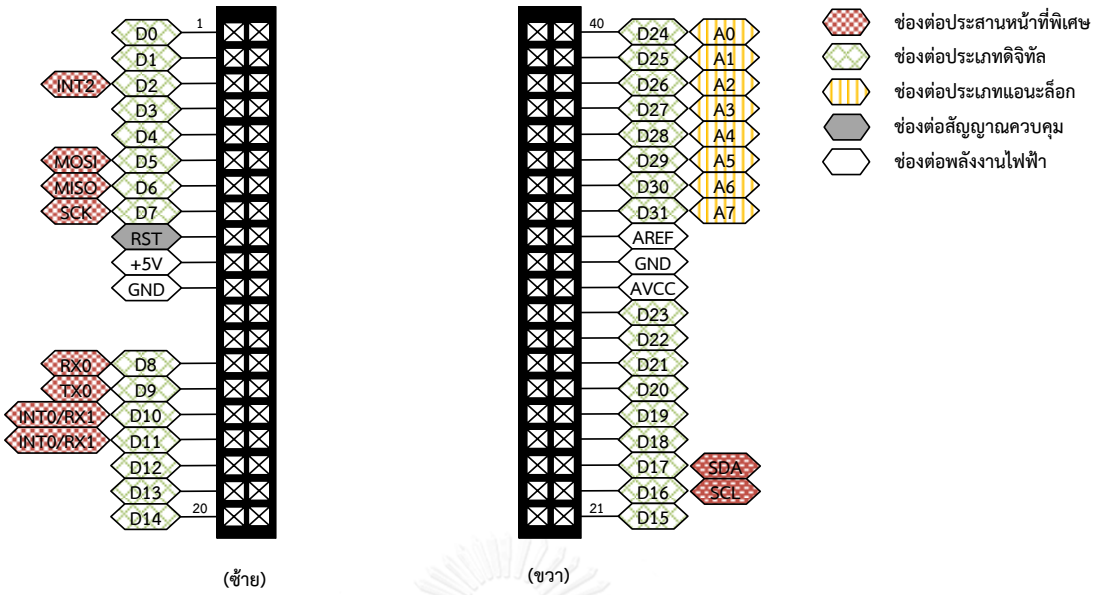
รูปที่ 5.5 โครงสร้างแผงวงจรของต้นแบบโนดเซนเซอร์หลักรุ่นที่ 2 ด้านหลัง

แผงวงจรหลักของต้นแบบโนดเซนเซอร์รุ่นที่ 2 มีขนาดกว้าง 70 มิลลิเมตร ยาว 125 มิลลิเมตร ดังแสดงในรูปที่ 5.6 แผงวงจรมีขนาดเล็กลงกว่ารุ่นก่อนหน้าและปรับปรุงการเชื่อมขาสัญญาณของไมโครคอนโทรลเลอร์ให้สามารถใช้งานได้สะดวกมากขึ้น



รูปที่ 5.6 แผงวงจรหลักรุ่นที่ 2

ช่องต่อแผงวงจรส่วนต่อขยายออกแบบให้เป็นช่องต่อตัวเมียแถวคู่ด้านละ 20 ช่องสัญญาณ ดังแสดงในรูปที่ 5.7 เพื่อให้สามารถเชื่อมต่ออุปกรณ์พร้อมกับการตรวจสอบโดยเครื่องมือตรวจวัดทางไฟฟ้าได้สะดวกมากขึ้น



รูปที่ 5.7 โครงสร้างช่องต่อแผงวงจรส่วนต่อขยายต้นแบบโนดเซนเซอร์รุ่นที่ 2

5.1.2 ข้อมูลด้านฮาร์ดแวร์

ต้นแบบโนดเซนเซอร์ที่พัฒนาขึ้นในงานวิจัยชิ้นนี้ได้รับการพัฒนาอ้างอิงกับแพลตฟอร์มอาดูอิโนและมีการติดตั้งเฟิร์มแวร์ Bootloader ที่มีชื่อรุ่นว่า Opti-boot [53] พัฒนาโดยนักพัฒนาชื่อ Maniacbug มาใช้งานกำหนดชื่อขาสัญญาณของไมโครคอนโทรลเลอร์และช่วยดาวน์โหลดโปรแกรมลงโปรแกรมจากคอมพิวเตอร์ตามรูปแบบการใช้งานในแพลตฟอร์มอาดูอิโน การกำหนดชื่อขาสัญญาณจะอยู่ในไฟล์ที่มีชื่อว่า pins_arduino.h และอนุญาตให้ผู้พัฒนาอื่นนำไปกำหนดชื่อใหม่ได้ตามความต้องการด้วย

การติดตั้ง Bootloader จะใช้เครื่องโปรแกรม AVR ISP ผ่านช่องต่อเครื่องโปรแกรม เมื่อติดตั้งเสร็จแล้วสามารถตรวจสอบการทำงานของ Bootloader ได้โดยให้สังเกตที่หลอดแอลอีดีแสดงสถานะ เมื่อไมโครคอนโทรลเลอร์ได้รับสัญญาณรีเซตและเริ่มทำงานหลอดแอลอีดีแสดงสถานะจะกระพริบเป็นจังหวะ 3 ครั้ง แล้วจะดับไปแสดงว่า Bootloader ติดตั้งได้ถูกต้องและทำงานเป็นปกติ การใช้งานต้นแบบโนดเซนเซอร์รุ่นที่ 2 มีรูปแบบที่เหมือนกับการใช้งานบอร์ดอาดูอิโนมาตรฐานทั่วไป แต่มีการเพิ่มวงจรฐานเวลาจริง วงจรเพิ่มแรงดันแบตเตอรี่ และ วงจรตรวจสอบแบตเตอรี่เข้ามาไว้ในแผงวงจรหลัก ทำให้การใช้งานมีความสะดวกกว่าการนำบอร์ดอาดูอิโนมาต่อวงจรเหล่านี้เพิ่มเข้าไปจากภายนอก รวมถึงลดความผิดพลาดในการเชื่อมต่อได้อีกด้วย การเรียกฟังก์ชันมาตรฐานของแพลตฟอร์มอาดูอิโน การเข้าถึงเรจิสเตอร์ของไมโครคอนโทรลเลอร์ยังสามารถทำได้ตามที่ผู้พัฒนาคุ้นเคย ข้อมูลที่ด้านเทคนิคที่สำคัญสำหรับการใช้งานต้นแบบโนดเซนเซอร์แสดงในตารางที่ 5.1

ตารางที่ 5.1 ข้อมูลด้านเทคนิคต้นแบบโนดเซนเซอร์รุ่นที่ 2

ไมโครคอนโทรลเลอร์	ATmega1284P-PU (PDIP-40)
ความถี่สัญญาณนาฬิกา	16 เมกะเฮิรตซ์
หน่วยความจำแบบแฟลช	128 กิโลไบต์ (Bootloader 1 กิโลไบต์)
แรม	16 กิโลไบต์
EEPROM	4 กิโลไบต์
ประเภทช่องต่อประสาน	• ช่องต่อดิจิทัลจำนวน 31 ช่อง • ช่องต่อแอนะล็อก 8 ช่อง • พอร์ตอนุกรม 2 ช่อง • บัสเอสพีไอ • บัส I^2C
การต่ออุปกรณ์ภายนอก	ช่องต่อแผงวงจรส่วนขยายขนาด 20 ช่องสัญญาณ 2 ชุด
ประเภทการสื่อสาร	• เครื่องรับส่งสัญญาณวิทยุคลื่นความถี่ 2.4 กิกะเฮิรตซ์ • เครื่องรับส่งสัญญาณวิทยุเทคโนโลยีซิกบี • เครื่องรับส่งสัญญาณวิทยุเทคโนโลยีบูลทูท
วงจรพิเศษ	• วงจรฐานเวลาจริงพร้อมแบตเตอรี่สำหรับสำรองข้อมูล • วงจรแปลงระดับแรงดันไฟฟ้า • วงจรตรวจวัดแบตเตอรี่
วิธีดาวน์โหลดโปรแกรม	• เครื่องโปรแกรม AVR ISP mkII • พอร์ตอนุกรมด้วยเฟิร์มแวร์ Bootloader
แหล่งพลังงาน	• แบตเตอรี่ลิเธียมไอออน 1 เซลล์ 3.7 โวลต์ • พอร์ตยูเอสบีจากคอมพิวเตอร์
การใช้พลังงาน	• โหมดการทำงานปกติ 28 มิลลิแอมแปร์ • โหมดนอนหลับ 20 ไมโครแอมแปร์

เมื่อพิจารณาข้อมูลด้านเทคนิคของต้นแบบโนดเซนเซอร์เปรียบเทียบกับบอร์ดอาดูอิโนมาตรฐาน และ Waspmote แล้วพบว่าการนำบอร์ดอาดูอิโนมาตรฐานมาใช้งานนั้นจะมีข้อจำกัดในด้านพลังงานเนื่องจากไม่ได้ออกแบบสำหรับการใช้พลังงานจากแบตเตอรี่ ในส่วนข้อจำกัดด้านวงจรที่จำเป็นต่อการใช้งานเป็นโนดเซนเซอร์ที่ขาดหายไป อาทิ วงจรฐานเวลาจริง วงจรเพิ่มแรงดันแบตเตอรี่ วงจรตรวจวัดแบตเตอรี่ การต่อวงจรเพิ่มจากภายนอกบอร์ดอาจพบปัญหาการเชื่อมต่อและไม่สะดวก

ต่อการนำไปใช้งาน นอกจากนี้บอร์ดอาดูอิโนมาตรฐาน ARUINO MEGA 2560 ที่มีหน่วยความจำขนาดใหญ่ แต่มีความซับซ้อนในการซ่อมแซมที่มากกว่าและราคาจำหน่ายค่อนข้างสูงจึงไม่เหมาะสำหรับนำมาใช้งานเป็นโนดเซนเซอร์สำหรับตรวจวัด เมื่อเปรียบเทียบกับโนดเซนเซอร์ Waspmote ที่กำลังได้รับความนิยมในขณะนี้พบว่าต้นแบบโนดเซนเซอร์สามารถบัสเอสพีไอเพื่อใช้งานเครื่องส่งสัญญาณวิทยุจากผู้ผลิตรายอื่นได้และยังสามารถใช้เครื่องรับส่งสัญญาณวิทยุ Xbee ได้เช่นเดียวกับ Waspmote อีกด้วย สำหรับการซ่อมแซมเมื่อเกิดความเสียหายกับหน่วยประมวลผลก็สามารถถอดเปลี่ยนไมโครคอนโทรลเลอร์และติดตั้งเฟิร์มแวร์ได้ด้วยตัวเอง การเปรียบเทียบโนเซนเซอร์ต้นแบบกับบอร์ดอาดูอิโนและ Waspmote แสดงในตารางที่ 5.2

ตารางที่ 5.2 การเปรียบเทียบข้อมูลด้านเทคนิคระหว่างต้นแบบโนดเซนเซอร์กับบอร์ดอาดูอิโนมาตรฐาน และ Waspmote

บอร์ด	**ต้นแบบ โนดเซนเซอร์**	**ARDUINO UNO**	**ARDUINO MEGA 2560**	**Waspmote**
ไมโครคอนโทรลเลอร์	ATmega1284P (8 bits)	ATmega328 (8 bits)	ATmega2560 (8 bits)	ATmega1281 (8 bits)
หน่วยความจำแฟลช (กิโลไบต์)	128	32	256	128
แรม (กิโลไบต์)	16	2	8	8
EEPROM (กิโลไบต์)	4	1	4	4
สัญญาณนาฬิกา (เมกะเฮิรตซ์)	16	16	16	8
ช่องต่อประสานพิเศษ				
• พอร์ตอนุกรม	√	√	√	√
• บัสเอสพีไอ	√	√	√	-
• บัส I^2C	√	√	√	√
วงจรพิเศษ				
• วงจรฐานเวลาจริง	√	-	-	√
• วงจรเพิ่มแรงดัน	√	-	-	√
• วงจรตรวจวัดแบตเตอรี่	√	-	-	√
การดาวน์โหลดโปรแกรม	USB/AVR ISP	USB/AVR ISP	USB/AVR ISP	USB

5.2 ส่วนต่อประสานโปรแกรมประยุกต์ Simple-API

ส่วนต่อประสานโปรแกรมประยุกต์ Simple-API พัฒนาขึ้นเพื่ออำนวยความสะดวกในการพัฒนาโปรแกรมประยุกต์สำหรับใช้งานโนดเซนเซอร์ มีรูปแบบการพัฒนาขึ้นเป็นคลาสแทนส่วนการทำงานของโนดเซนเซอร์ทำหน้าที่ควบคุมสั่งงานอุปกรณ์โดยใช้มอดูลแทนอุปกรณ์จริง พารามิเตอร์แทนค่าที่ต้องการปรับตั้งในการทำงานและฟังก์ชันสมาชิกของคลาสสำหรับใช้สั่งงาน ผลการพัฒนาและเรียกใช้ Simple-API ในงานวิทยานิพนธ์นี้จะนำมาเปรียบเทียบกับรูปแบบและขั้นตอนการพัฒนาโปรแกรมประยุกต์สำหรับงานเครือข่ายเซนเซอร์ไร้สายโดยใช้บอร์ดอาดูอิโนและ Waspmote เนื่องจากฮาร์ดแวร์เลือกใช้งานไมโครคอนโทรลเลอร์ตระกูล AVR รุ่น 8 บิต ที่มีความใกล้เคียงกันตลอดจนรูปแบบการพัฒนาซอฟต์แวร์ที่มีการเตรียมตัวอย่างการเชื่อมต่ออุปกรณ์ คลังโปรแกรมขับอุปกรณ์ ขั้นตอนการพัฒนาที่มีลักษณะที่คล้ายกันอีกด้วย ผลการพัฒนา Simple-API ทั้งหมดมีรายละเอียดดังต่อไปนี้

5.2.1 การสร้างและจัดกลุ่มฟังก์ชันการทำงาน

การออกแบบฟังก์ชันสำหรับเข้าถึงมอดูลและพารามิเตอร์จะใช้ฟังก์ชันสาธารณะที่เป็นสมาชิกของคลาสสามารถแบ่งได้ทั้งหมด 3 กลุ่ม ได้แก่ กลุ่มฟังก์ชันสำหรับเริ่มการทำงานอุปกรณ์ กลุ่มฟังก์ชันสำหรับการปรับแต่งพารามิเตอร์ของอุปกรณ์ และ กลุ่มฟังก์ชันสำหรับควบคุมสั่งงานอุปกรณ์ การเรียกใช้งานอุปกรณ์สำหรับเริ่มทำงานจะต้องเรียกใช้ฟังก์ชันในกลุ่มแรกก่อนเสมอเพื่อสร้างการเชื่อมต่อไปยังอุปกรณ์และกำหนดค่าที่จำเป็นในการใช้งาน หากต้องการปรับตั้งค่าหรืออ่านค่าของอุปกรณ์จะเรียกใช้ฟังก์ชันในกลุ่มที่สอง และ การสั่งงานอุปกรณ์เพื่อตรวจวัดหรือรับส่งข้อมูลจะเรียกฟังก์ชันในกลุ่มสุดท้าย การเรียกใช้งานกลุ่มฟังก์ชันแรกและกลุ่มฟังก์ชันที่สองจะมีความคล้ายกันทั้ง 4 คลาส ผู้ใช้สามารถศึกษาวิธีการเรียกใช้งานเพียงคลาสเดียวก็จะสามารถเรียกใช้งานคำสั่งในคลาสอื่นได้ ฟังก์ชันสมาชิกแบบสาธารณะของแต่ละคลาสมีการจัดแบ่งกลุ่มดังแสดงในตารางที่ 5.3

ตารางที่ 5.3 รายชื่อฟังก์ชันสมาชิกแบบสาธารณะของแต่ละคลาส

คลาส	ฟังก์ชันสั่งเริ่มการทำงาน	ฟังก์ชันจัดการพารามิเตอร์	ฟังก์ชันสั่งงานอุปกรณ์

ProcessingSystemC	begin();	writeParameter(module, paramName, *value, valueLength);
		readParameter(module, paramName, *value, valueLength);
			การจัดบันทึกข้อมูลผู้ใช้ลง EEPROM
			writeUserData(page, *value, valueLength);
			readUserData(page, *value, length);
			การควบคุมหลอดแอลอีดีแสดงสถานะ
			onBoardLedOn();
			onBoardLedOff();
			onBoardLedBlink(times);
			การทำงานไมโครคอนโทรลเลอร์
			sleepNow();
			resetNow();

SensorAndActuatorC	begin(module, interfaceName);	writeParameter(module, paramName, *value, valueLength);
		readParameter(module, paramName, *value, valueLength);
			การอ่านค่าการตรวจวัดจากเซนเซอร์
			readSensorData(module, *value, valueLength);
			การปรับตั้งค่ามาตรฐาน
			calibrateSensor(*ptrDisplay, module);

WirelessCommunication SystemC	begin(module, interfaceName);	writeParameter(module, paramName, *value, valueLength);
		readParameter(module, paramName, *value, valueLength);
			การส่งและรับข้อมูลผ่านเครื่องส่งสัญญาณวิทยุ
			sendData(module, *destination, *msg, msgLength);
			readData(module, *msg, *msgType, msgLength);

PowerControlSystemC	begin();	writeParameter(module, paramName, *value, valueLength);
		readParameter(module, paramName, *value, valueLength);
			การควบคุมการใช้พลังงาน
			enablePower(module);
			disablePower(module);

รายละเอียดการใช้งานฟังก์ชัน มอดูล และพารามิเตอร์อยู่ในภาคผนวก

เมื่อพิจารณาคำสั่งเริ่มต้นสำหรับพัฒนาโปรแกรมประยุกต์ของโนดเซนเซอร์ ผู้ใช้จะต้องเข้าไปกำหนดค่าให้กับไมโครคอนโทรลเลอร์ วงจรฐานเวลาจริง รวมถึงหลอดแอลอีดีสำหรับการแสดงสถานะ การพัฒนาโดยใช้ฟังก์ชัน begin(); ของคลาส ProcessingSystemC จะช่วยให้ผู้ใช้ไม่ต้องเข้าไปจัดการรีจิสเตอร์และการเขียนโปรแกรมในระดับล่าง ฟังก์ชัน begin(); จะทำการจัดกลุ่มฟังก์ชันที่จำเป็นต่อการใช้งานเบื้องต้นไว้ให้โดยไม่ต้องเข้าไปปรับแต่งเพิ่มเติม หากเปรียบเทียบกับการพัฒนาโดยใช้อาดูอิโนผู้ใช้งานจำเป็นต้องเขียนคำสั่งสำหรับการเชื่อมต่อวงจรฐานเวลาจริง การตั้งค่าการอินเทอร์รัพท์ การกำหนดโหมดการทำงาน รวมถึงการกำหนดค่าเพื่อใช้งานหลอดแอลอีดีแสดงสถานะ ซึ่งขั้นตอนที่กล่าวมาเป็นการเขียนโปรแกรมระดับล่างที่เกี่ยวข้องกับเรจิสเตอร์และการทำงานของอินเทอร์รัพท์ที่มีความซับซ้อนสำหรับผู้เริ่มไม่เคยใช้งานมาก่อน ในส่วนการพัฒนาโดยใช้งาน Waspmote นั้นแม้ว่าผู้ผลิตได้เตรียมคลังโปรแกรมไว้ให้ใช้งานสำหรับอุปกรณ์ที่ติดตั้งมาบนบอร์ดแล้ว แต่ผู้ใช้ยังคงต้องศึกษาว่าจะต้องจัดการกับอุปกรณ์ตัวใดบ้าง มีลำดับการจัดการอุปกรณ์อย่างไร รวมถึงวิธีการเรียกใช้งานอุปกรณ์แต่ละตัวที่แตกต่างกัน การเปรียบเทียบการเขียนคำสั่งที่ต้องเรียกใช้งานในการเริ่มพัฒนาโปรแกรมประยุกต์แสดงในตารางที่ 5.4

ตารางที่ 5.4 การเปรียบเทียบคำสั่งที่ใช้ในการเริ่มทำงานของโนดเซนเซอร์

Simple-API	Waspmote	อาดูอิโน

ProcessingObj.begin();	// คำสั่งเริ่มการทำงานวงจรฐานเวลาจริงและอินเทอร์รัพท์สำหรับตั้งปลุก //
	RTC.setMode(RTC_ON);
	RTC.ON();
		// คำสั่งสร้างการเชื่อมต่อและเริ่มการทำงานวงจรฐานเวลาจริง //
		Wire.begin();
		WriteByte();
		...
		Wire.endTransmission();
		// คำสั่งตั้งค่าและการบริการอินเทอร์รัพท์ //
		begin(FALLING_EDGE);
		attatchInterrupt(2);
		interruptISR_vect();
		// คำสั่งตั้งค่าการทำงานโหมดนอนหลับ //
		set_sleep_mode(Mode);

	// คำสั่งตั้งค่าหลอดแอลอีดีแสดงสถาน //
	Utils.setLED(LED1, LED_OFF);
	Utils.setLED(LED0, LED_OFF);
		// คำสั่งตั้งค่าหลอดแอลอีดีแสดงสถานะ //
		pinMode(ledPin, OUTPUT);
		digitalWrite(ledPin, HIGH);

5.2.2 การลดจำนวนฟังก์ชันและยืดหยุ่นกับอุปกรณ์

การลดจำนวนฟังก์ชันที่ต้องศึกษาสำหรับใช้งานอุปกรณ์ใน Simple-API ใช้สัญลักษณ์มอดูลและพารามิเตอร์ ร่วมกับฟังก์ชันที่ใช้สำหรับจัดการเพียง 2 ฟังก์ชัน ได้แก่ writeParameter(); และ readParameter(); สำหรับการปรับค่าการใช้งานในทุก ๆ คลาส เมื่อเปรียบเทียบรูปแบบการใช้งานโปรแกรมขับอุปกรณ์หรือคลังโปรแกรมของแพลตฟอร์มอาดูอิโนและ Waspmote นั้นพบการฟังก์ชันที่ใช้งานมีจำนวนมากและมีรูปแบบที่แตกต่างกันโดยเฉพาะในส่วนการปรับตั้งค่าเครื่องรับส่งสัญญาณวิทยุหรือเซนเซอร์ ผู้ใช้อาจสับสนอาจทำให้เรียกใช้งานไม่ครบตามความต้องการได้ การใช้งาน Simple-API จะช่วยลดจำนวนฟังก์ชันจำนวนมากที่ใช้สำหรับการปรับตั้งค่าของอุปกรณ์ได้โดยเมื่อต้องการปรับตั้งค่าจะใช้ฟังก์ชัน writeParameter(); และการอ่านค่าพารามิเตอร์จะใช้ฟังก์ชัน readParameter(); ซึ่งสามารถเรียนรู้ได้ง่ายกว่า การเปรียบเทียบระหว่างการเรียกใช้ Simple-API และโปรแกรมขับอุปกรณ์โดยตรงในการปรับตั้งค่าเครื่องรับส่งสัญญาณวิทยุและการอ่านค่าข้อมูลแบตเตอรี่ของโนดเซนเซอร์แสดงในตารางที่ 5.5

การใช้งาน Simple-API ในโปรแกรมประยุกต์จะมีความยืดหยุ่นต่อการเปลี่ยนแปลงอุปกรณ์ เนื่องจาการปรับตั้งค่าหรือการสั่งงานจะใช้สัญญาลักษณ์มอดูลแทนอุปกรณ์จริงและคำสั่งที่ใช้งานสามารถใช้งานร่วมกับมอดูลอื่น ๆ ได้ภายในคลาสโดยไม่มีฟังก์ชันเฉพาะสำหรับอุปกรณ์เพียงตัวใดตัวหนึ่ง เมื่อเปรียบเทียบกับการเขียนคำสั่งเพื่อตรวจวัดจากเซนเซอร์ดังแสดงในตารางที่ 5.6 การเรียกใช้เซนเซอร์เพื่อตรวจวัดค่าพีเอชแล้วทำการเปลี่ยนไปใช้งานเซนเซอร์ตรวจวัดค่าออกซิเจนที่ละลายในน้ำ พบว่าการเรียกใช้งาน Simple-API โดยใช้คำสั่ง readSensorData(); มีการเปลี่ยนอาร์กิวเมนต์ตัวแรกเพียงตัวเดียวโดยเปลี่ยนจาก MODULE_ATLAS_PH ไปเป็น MODULE_ATLAS_DO ซึ่งมีความง่ายต่อการใช้งาน โดยไม่ต้องปรับเปลี่ยนโปรแกรมประยุกต์ผู้ใช้งานเลย เมื่อเปรียบเทียบกับการใช้งานอาดูอิโนและ Waspmote ผู้ใช้งานจะต้องแก้ไขโปรแกรมประยุกต์ใหม่โดยเริ่มจากการลบวัตถุตัวเดิมที่ใช้ภายในโปรแกรม สร้างวัตถุใหม่สำหรับอุปกรณ์ที่ทำการเปลี่ยนใหม่และเขียนคำสั่งเพื่อเรียกใช้งานซึ่งอาจมีรูปแบบที่คล้ายกับการใช้งานอุปกรณ์ตัวเดิมหรือมีรูปแบบที่แตกต่างออกไป การแก้ไขดังกล่าวอาจทำให้โปรแกรมประยุกต์ที่พัฒนาเสร็จแล้วมีข้อผิดพลาดเกิดขึ้นได้ง่ายและอาจต้องเสียเวลาในการพัฒนาที่มากขึ้นกว่าการใช้งาน Simple-API สำหรับการพัฒนาโปรแกรมประยุกต์

ตารางที่ 5.5 ฟังก์ชันตั้งค่าอุปกรณ์ของ Simple-API เมื่อเทียบกับ Waspmote และ อาดูอิโน

	การเรียกใช้งาน

Simple-API	// การตั้งค่าช่องสัญญาณการสื่อสาร //
	WirelessCommObj.**writeParameter**(MODULE_RADIO_24, **P_CHANNEL**,
	&channel, sizeof(channel));
	
	// การตั้งค่าอัตราเร็วในการสื่อสาร //
	WirelessCommObj.**writeParameter**(MODULE_RADIO_24, **P_DATA_RATE**,
	&rate, sizeof(rate));
	
	// การอ่านค่าแรงดันไฟฟ้าของแบตเตอรี่ //
	PowerObj.**readParameter**(MODULE_BATTERY, **PW_BAT_VOLTAGE**,
	&voltage, sizeof(voltage));
	
	// การอ่าค่าความจุคงเหลือของแบตเตอรี่ //
	PowerObj.**readParameter**(MODULE_BATTERY, **PW_BAT_CAPACITY**, &capacity,
	sizeof(capacity));

Waspmote และ อาดูอิโน	// การตั้งค่าช่องสัญญาณการสื่อสาร //
	nRF24Module.setChannel(channel);
	
	// การตั้งค่าอัตราเร็วในการสื่อสาร //
	nRF24Module.setDataRate(rate);
	
	// การอ่านค่าแรงดันไฟฟ้าของแบตเตอรี่ //
	float battery_vol;
	PWR.getBatteryVol(&battery_vol);
	
	// การอ่าค่าความจุคงเหลือของแบตเตอรี่ //
	battery_level = PWR.getBatteryLevel();

ตารางที่ 5.6 รูปแบบการเรียกใช้งานอุปกรณ์ของ Simple-API เมื่อเทียบกับ Waspmote และ อาดูอิโน

	การเรียกใช้เซนเซอร์ตรวจวัดค่าพีเอช	การเรียกใช้เซนเซอร์ตรวจวัดค่าออกซิเจนที่ละลายในน้ำ

Simple-API	// การสร้างวัตถุส่วนตรวจวัดและสั่งการ //
	SensorAndActuatorSystemC SensorObj;
	
	// ตัวแปรสำหรับเก็บค่า //
	float value = 0.0;
	
	// คำสั่งเริ่มการทำงานของเซนเซอร์ //
	SensorObj.begin(MODULE_ATLAS_PH,HW_SERIAL_1);
	
	// คำสั่งตรวจวัด //
	SensorObj.readSensorData(MODULE_ATLAS_PH, *value, sizeof(value));
		// การสร้างวัตถุส่วนตรวจวัดและสั่งการ //
		SensorAndActuatorSystemC SensorObj;
		
		// ตัวแปรสำหรับเก็บค่า //
		float value = 0.0;
		
		// คำสั่งเริ่มการทำงานของเซนเซอร์ //
		SensorObj.begin(MODULE_ATLAS_DO, HW_SERIAL_1);
		
		// คำสั่งตรวจวัด //
		SensorObj.readSensorData(MODULE_ATLAS_DO, *value, sizeof(value));

Waspmote หรือ อาดูอิโน	// การสร้างวัตถุของเซนเซอร์ //
	AtlasPHSensor pHSensor;
	
	// ตัวแปรสำหรับเก็บค่า //
	float value = 0.0;
	
	// คำสั่งเริ่มการทำงานของเซนเซอร์ //
	pHSensor.initial();
	
	// คำสั่งตรวจวัด //
	value = pHSensor.read();
		// การสร้างวัตถุของเซนเซอร์ //
		AtlasDOSensor DOSensor;
		
		// ตัวแปรสำหรับเก็บค่า //
		float value = 0.0;
		
		// คำสั่งเริ่มการทำงานของเซนเซอร์ //
		DOSensor.initial();
		
		// คำสั่งตรวจวัด //
		DOSensor.read(*value);

5.2.3 การป้องการความผิดพลาดการเรียกใช้งานและการแจ้งผลการทำงาน

การทำงานที่ผิดพลาดในขั้นตอนการพัฒนาโปรแกรมประยุกต์สำหรับโนดเซนเซอร์สามารถเกิดได้เป็นปกติ เพื่อป้องกันการทำงานที่ผิดพลาดของโนดเซนเซอร์ Simple-API ถูกออกแบบให้ป้องกันการกำหนดค่าหรือเรียกใช้งานอุปกรณ์ที่ไม่สามารถทำงานได้ หากผู้ใช้เผลอกำหนดค่าที่ไม่สามารถทำงานได้ Simple-API จะกำหนดค่าเริ่มต้นที่เหมาะสมกับการทำงานให้กับอุปกรณ์นั้นแทน เพื่อป้องกันการทำงานที่ผิดพลาดร้ายแรงและแจ้งผลการเรียกใช้งานผ่านทางรหัสตัวเลขที่ใช้แทนความผิดพลาดที่เกิดขึ้นดังแสดงในตารางที่ 5.7 ซึ่งแตกต่างจากการคืนค่ากลับของการเรียกใช้โปรแกรมขับอุปกรณ์ของ Waspmote เนื่องจากจะมีการคืนค่ากลับเพียงสองค่า คือ 0 หมายความว่าทำงานได้สำเร็จ และ 1 หมายความว่าทำงานไม่สำเร็จ โดยไม่ได้ให้รายละเอียดของการทำงานเท่ากับ Simple-API ในส่วนของอาดูอิโนผู้ใช้จะต้องเป็นผู้ตรวจสอบการทำงานที่ผิดพลาดเองในขั้นตอนการเชื่อมต่อและการกำหนดค่าเนื่องจากจะเขียนคำสั่งการเชื่อมต่อและโปรแกรมขับอุปกรณ์ไว้คนละคลาสแยกจากกัน

นอกจากการรายงานค่าเป็นรหัสตัวเลขแล้ว Simple-API ยังป้องกันความผิดพลาดระหว่างทำงานหากการสื่อสารกับอุปกรณ์ตัวนั้นผิดพลาดหรือขาดหายไป สำหรับผู้เริ่มพัฒนาโปรแกรมประยุกต์สำหรับไมโครคอนโทรลเลอร์จะไม่ได้คำนึงถึงเหตุการณ์เช่นนี้มากนัก เช่น การสื่อสารกับอุปกรณ์ที่กำหนดส่วนหัว (header) และส่วนท้าย (footer) ของการส่วข้อมูล หากผู้ใช้ไม่ระวังโดยเขียนคำสั่งวนรอการอ่านค่าส่วนท้ายเพียงอย่างเท่านั้น หากข้อมูลส่วนท้ายนี้หายไปจะทำให้โปรแกรมไม่สามารถออกมาจากการทำงานส่วนนั้นได้เลย Simple-API จึงได้เพิ่มชุดคำสั่งภายในคลาสให้มีกระบวนการส่งคำสั่งงานเดิมซ้ำ (retry) และรอการตอบกลับภายในเวลาที่กำหนด (time out) แต่หากการตอบกลับไม่ถูกต้องหรือไม่ตอบกลับภายระยะเวลาที่กำหนดก็จะหยุดจากการทำงานในส่วนนั้นแล้วทำการแจ้งผลการทำงานให้ผู้ใช้ทราบ การป้องกันดังกล่าวจะช่วยไม่ให้โนดเซนเซอร์เข้าไปทำงานค้างในคำสั่งงานนั้นโดยไม่สามารถออกมาทำงานในส่วนอื่นได้

ตารางที่ 5.7 ผลการทำงานที่เป็นไปได้ของ Simple-API เมื่อเทียบกับ Waspmote และ อาดูอิโน

	การเรียกใช้งาน	ผลการทำงานที่เป็นไปได้
Simple-API	WirelessCommObj.writeParameter(MODULE_RADIO_24, P_CHANNEL, &channel, sizeof(channel));	0 - ทำงานสำเร็จ 4 - ค่าหรือขนาดพารามิเตอร์ไม่ถูกต้อง -1 - ทำงานไม่สำเร็จอุปกรณ์เชื่อมต่อไม่ตอบสนองในเวลาที่กำหนด -2 - ทำงานไม่สำเร็จการเชื่อมต่อผิดพลาด
Waspmote	nRF24Module.setChannel(channel);	0 – ทำงานสำเร็จ 1 - ผลการทำงานเป็นอย่างอื่น

อาดูอิโน	csn_level(LOW);
	SPI.transfer(addr);
	for (uint8_t i=0; i<length; i++) {
	SPI.transfer(*ptrValue);
	ptrValue++;
	}
	csn_level(HIGH);
		ไม่มี ผู้ใช้งานจะต้องจัดการความผิดพลาดด้วยตัวเอง

5.3 การทดสอบการทำงานของต้นแบบโนดเซนเซอร์และ Simple-API

การทดสอบการทำงานของต้นแบบโนดเซนเซอร์จะเขียนคำสั่งเพื่อทดสอบการทำงานส่วนประมวลผล ส่วนตรวจวัดและสั่งการ ส่วนสื่อสารไร้สาย และส่วนควบคุมพลังงานและแบตเตอรี่ร่วมกับการเรียกใช้งานส่วนต่อประสานโปรแกรมประยุกต์ Simple-API โดยมีการทดสอบดังต่อไปนี้

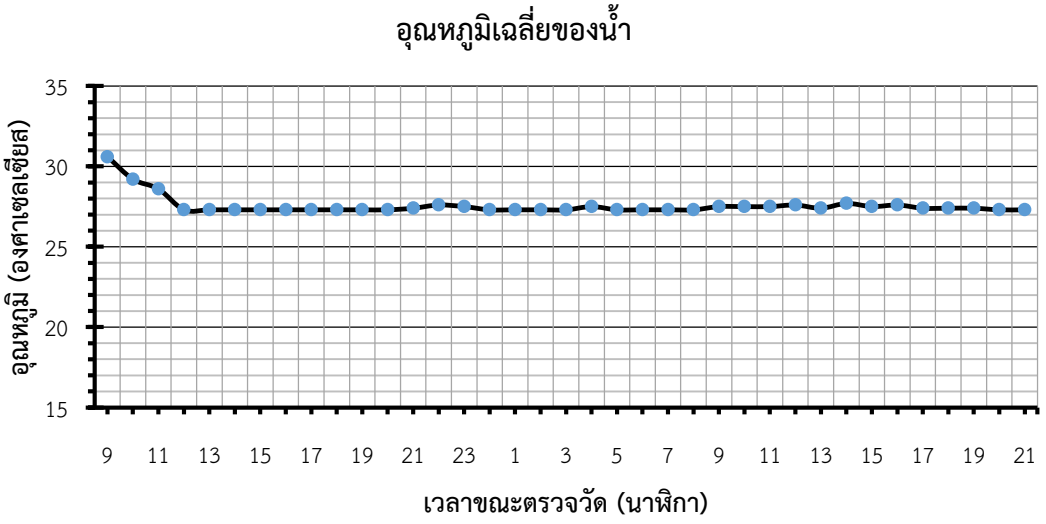
5.3.1 การสั่งงานอุปกรณ์บนโนดเซนเซอร์

การทดสอบการสั่งงานโนดเซนเซอร์ทำในห้องปฏิบัติการโดยการเขียนโปรแกรมเรียกใช้งานเซนเซอร์ตรวจวัดอุณหภูมิและตรวจวัดข้อมูลพลังงานไฟฟ้าจากแบตเตอรี่ร่วมกับทำงานในโหมดนอนหลับ ซึ่งมีช่วงเวลาการถูกปลุกให้ตื่นขึ้นมาทำงานในทุก 10 นาที ทั้งหมด 36 ชั่วโมง ผลการตรวจวัดอุณหภูมิและข้อมูลแบตเตอรี่มีรายละเอียดดังต่อไปนี้

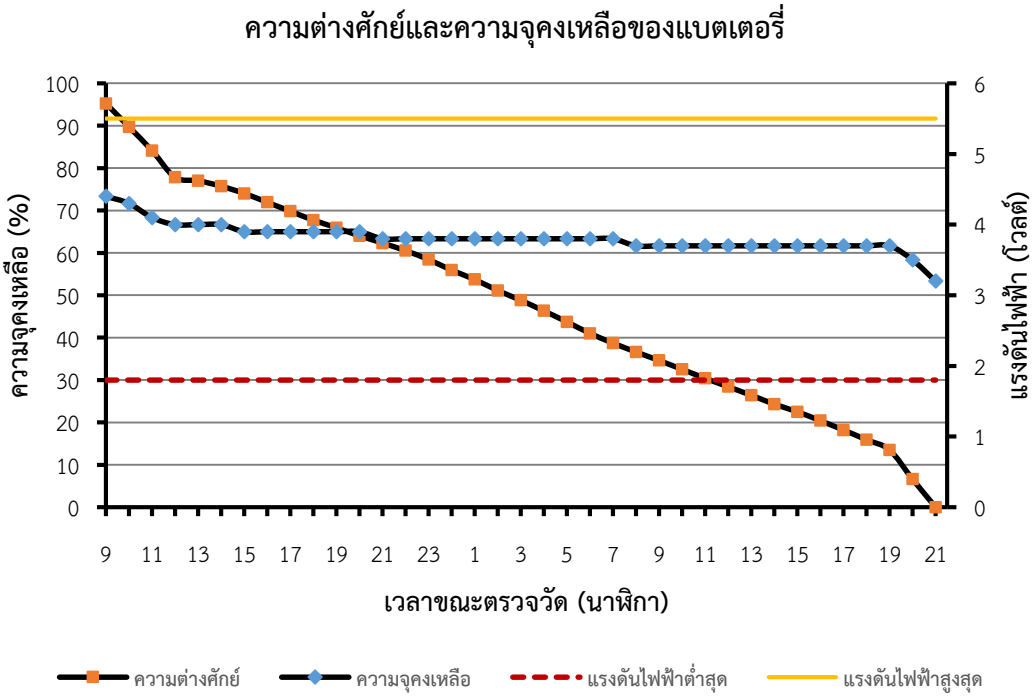
ข้อมูลอุณหภูมิที่แสดงในรูปที่ 5.8 เป็นการตรวจวัดอุณหภูมิโดยจุ่มเซนเซอร์ลงในตัวอย่างน้ำที่นำมาทดสอบ ในช่วงแรกของการตรวจวัดมีอุณหภูมิประมาณ 31 องศาเซลเซียสจากนั้นประมาณ 3 ชั่วโมง (เวลาประมาณ 12 นาฬิกา) อุณหภูมิของน้ำมีค่าคงที่ประมาณ 27 องศาเซลเซียส จากการเปิดเครื่องปรับอากาศภายในห้องปฏิบัติการ ต่อมาในช่วงท้ายของการตรวจวัด (เวลาประมาณ 12 – 17 นาฬิกา ของวันที่สอง) อุณหภูมิของน้ำที่ตรวจวัดได้มีค่าสูงขึ้นเล็กน้อยสาเหตุอาจมาจากมีผู้ใช้จำนวน

มากเข้าออกในห้องปฏิบัติการในตลอดช่วงบ่ายทำให้อุณหภูมิของห้องสูงขึ้นและช่วงสุดท้ายของการทดลอง (เวลาประมาณ 21 นาฬิกา ของวันที่สอง) อุณหภูมิกลับมามีค่าประมาณ 27 องศาเซลเซียสอีกครั้ง

ข้อมูลแรงดันไฟฟ้าและความจุคงเหลือของแบตเตอรี่แสดงในรูปที่ 5.9 แบตเตอรี่ที่นำมาใช้ทดลองเป็นประเภทลิเธียมไอออนขนาดความจุ 400 มิลลิแอมแปร์ชั่วโมง แรงดันไฟฟ้ามีค่าเริ่มเริ่มต้นประมาณ 4.4 โวลต์ เมื่อพิจารณาข้อมูลที่ได้จากการตรวจวัดแบตเตอรี่พบว่าค่าแรงดันไฟฟ้าจะมีค่าประมาณ 3.9 – 3.7 โวลต์ ตลอดการใช้งานแม้ว่าค่าความจุคงเหลือจะลดลงไปเรื่อย ๆ ซึ่งถือว่าเป็นข้อดีของการใช้งานแบตเตอรี่ประเภทนี้ ในช่วงสุดท้ายของการทดลอง (ในช่วงเวลาประมาณ 18 นาฬิกาของวันที่สอง) แรงดันไฟฟ้าจะมีค่าลดลงจนเหลือประมาณ 3.3 โวลต์และความจุไฟฟ้าประมาณ 10% แม้ว่าต้นแบบโนดเซนเซอร์จะสามารถใช้งานได้จนแรงดันไฟฟ้าเหลือเพียง (1.8 โวลต์) แต่ควรจะหยุดใช้งานและนำแบตเตอรี่ไปประจุไฟซ้ำเพื่อป้องกันความเสียหายที่จะเกิดขึ้นกับแบตเตอรี่



รูปที่ 5.8 ข้อมูลอุณหภูมิเฉลี่ยของน้ำที่ตรวจวัดตลอดช่วงการทดลอง



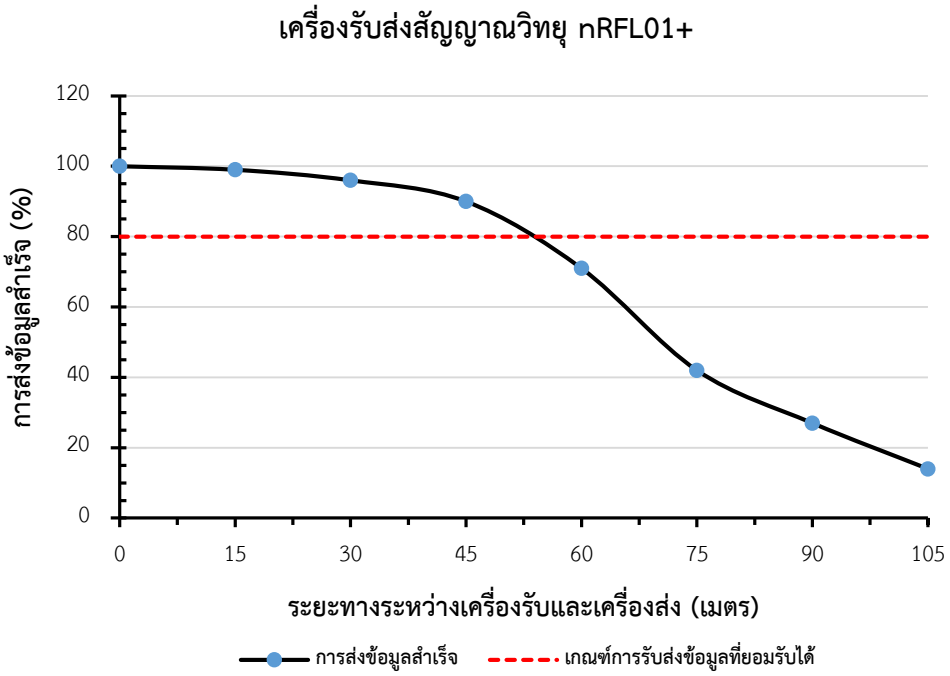
รูปที่ 5.9 ข้อมูลแรงดันไฟฟ้าและความจุคงเหลือเฉลี่ยของแบตเตอรี่ตลอดช่วงการทดลอง

การทดสอบการสั่งงานของโนดเซนเซอร์พบว่าสามารถทำงานตรวจวัด การทำงานในโหมดนอนหลับและปลุกขึ้นมาทำงานได้ปกติโดยไม่มีการทำงานค้างตลอดช่วงการทดลองทั้ง 36 ชั่วโมง ให้ค่าการตรวจวัดอุณหภูมิและข้อมูลพลังงานจากแบตเตอรี่ได้อย่างถูกต้องตามความเป็นจริง

5.3.2 การส่งข้อมูลผ่านเครื่องรับส่งสัญญาณวิทยุ

การทดสอบรับส่งข้อมูลของเครื่องรับส่งสัญญาณวิทยุจะทำในพื้นที่เปิดโล่ง ติดตั้งเครื่องส่งในแนวระนาบเดียวกัน ระยะห่างระหว่างโนดเซนเซอร์ที่ใช้ในการสื่อสารประมาณ 120 เมตร ใช้กำลังในการส่งสัญญาณสูงสุดของเครื่องรับส่งสัญญาณวิทยุ โดยตั้งเกณฑ์การรับส่งข้อมูลที่มีประสิทธิภาพเมื่อการส่งข้อมูลมีความสำเร็จประมาณ 80 เปอร์เซ็นต์ ผลการทดสอบการทำงานมีรายละเอียดดังนี้

เครื่องรับส่งสัญญาณวิทยุ nRFL01+ ให้ผลการทดสอบดังแสดงในรูปที่ 5.10 การทดสอบทำโดยการเขียนคำสั่งให้ส่งข้อมูลด้วยกำลังส่งสูงสุด 0 dBm ข้อมูลมีขนาด 32 ไบต์ ในส่วนเครื่องรับจะเขียนคำสั่งรับข้อมูลและนับข้อมูลที่รับได้และมีความถูกต้อง ทำการทดสอบซ้ำทั้งหมด 1,000 รอบต่อการเปลี่ยนระยะส่งสัญญาณทุก ๆ 15 เมตร ผลการทดสอบพบว่ามอดูล nRFL01+ จะสามารถสื่อสารได้ในระยะห่างไม่เกิน 55 เมตร จากนั้นความถูกต้องในการสื่อสารจะลดลงตามลำดับ เนื่องมาจากเครื่องรับส่งสัญญาณ nRFL01+ ใช้เสาอากาศที่สร้างบนแผงวงจรไม่สามารถใช้เสาอากาศจากภายนอกได้ทำให้สื่อสารได้ในระยะที่ไม่ไกลมากนัก

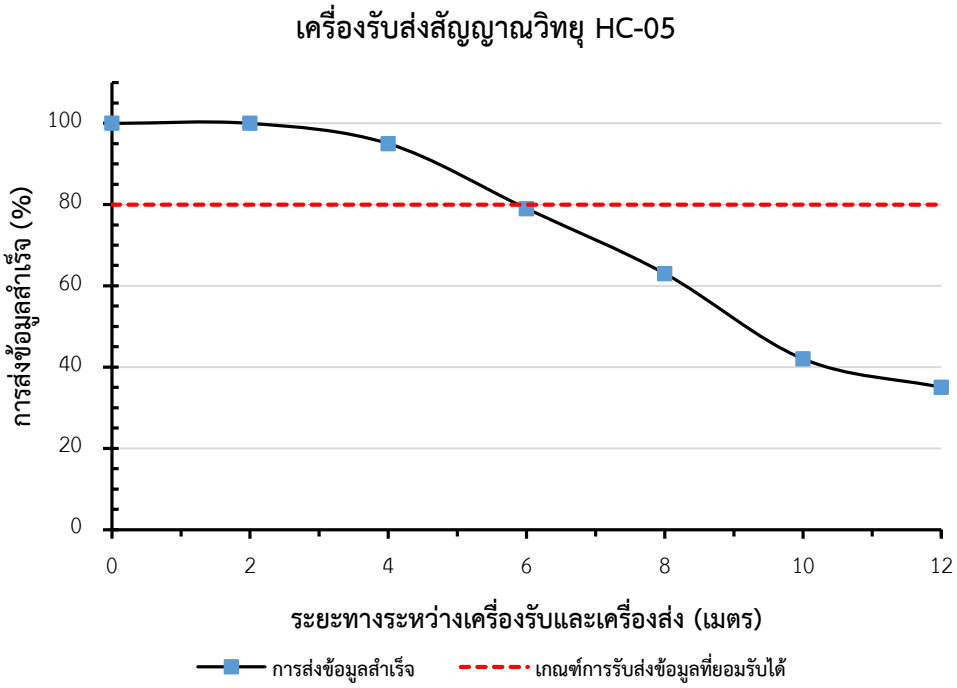


รูปที่ 5.10 ผลการทดสอบการส่งข้อมูลของเครื่องรับส่งสัญญาณวิทยุ nRFL01+

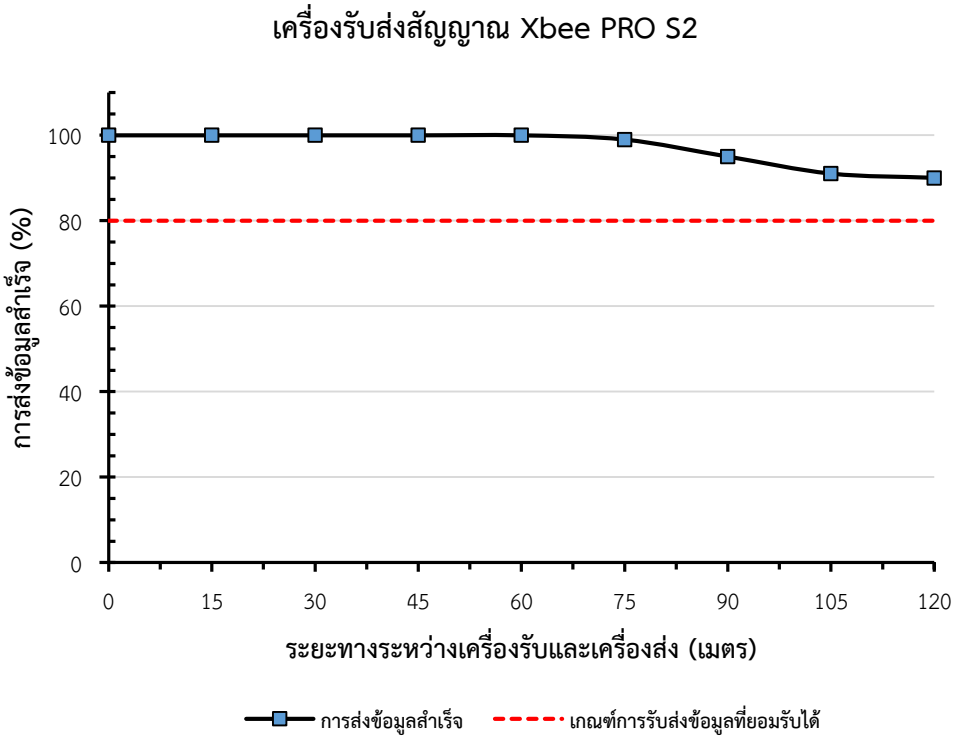
เครื่องรับส่งสัญญาณวิทยุ HC-05 ให้ผลการทดสอบดังแสดงแสดงในรูปที่ 5.11 การทดสอบทำโดยการจับคู่กับเครื่องคอมพิวเตอร์ที่ติดตั้งเครื่องรับส่งสัญญาณวิทยุบูลทูทก่อนทำการทดลองรับส่งข้อมูล โดยจะทำการส่งข้อมูลจากโนดเซนเซอร์ขนาด 10 ไบต์ ด้วยรหัสแอสกีไปยังคอมพิวเตอร์และนับข้อมูลที่ได้รับถูกต้องฝั่งคอมพิวเตอร์ทำการทดสอบซ้ำทั้งหมด 1,000 รอบต่อการเปลี่ยนระยะส่งสัญญาณทุก ๆ 2 เมตร ผลการทดสอบพบว่าเครื่องรับส่งสัญญาณวิทยุ HC-05 จะสามารถสื่อสารกับคอมพิวเตอร์ได้ดีใน ระยะห่างไม่เกิน 6 เมตร จากนั้นความถูกต้องของการส่งสัญญาณจะลดลง เนื่องจากเครื่องรับส่งสัญญาณวิทยุบูลทูท HC-05 มีการใช้เสาอากาศที่ติดตั้งบนแผงวงจรเช่นเดียวกับเครื่องรับส่งสัญญาณวิทยุ nRFL01+ ซึ่งอาจสื่อสารได้ในระยะไม่ไกลมากนักและเทคโนโลยีบูลทูทไม่ได้ออกแบบให้ใช้งานในระยะการส่งสัญญาณที่ไกล

เครื่องรับส่งสัญญาณวิทยุ Xbee PRO Zigbee S2 ให้ผลการทดสอบดังแสดงในรูปที่ 5.12 การทดสอบทำโดยการเขียนคำสั่งให้ส่งข้อมูลด้วยกำลังสูงสุด ข้อมูลมีขนาด 16 ไบต์ ในส่วนเครื่องรับจะเขียนคำสั่งรับข้อมูลและนับข้อมูลที่รับได้และมีความถูกต้อง ทำการทดสอบซ้ำทั้งหมด 1,000 รอบต่อการเปลี่ยนระยะส่งสัญญาณทุก ๆ 15 เมตร เครื่องรับส่งสัญญาณวิทยุ Xbee PRO Zigbee S2 สามารถสื่อสารได้ในระยะค่อนข้างไกลเนื่องจากมีกำลังส่งที่สูงกว่าเครื่องรับสัญญาณทั้งสอง 2 เครื่องที่กล่าวมาข้างต้น สามารถติดตั้งเสาอากาศภายนอกได้ เครื่องรับส่งสัญญาณวิทยุ Xbee PRO Zigbee

S2 สามารถรับส่งสัญญาณได้ดีที่ระยะ 120 เมตร ซึ่งเป็นค่าสูงสุดสำหรับระยะห่างระหว่างโนดเซนเซอร์ที่ใช้เป็นเกณฑ์กำหนดในการทดสอบนี้อีกด้วย



รูปที่ 5.11 ผลการทดสอบการส่งข้อมูลของเครื่องรับส่งสัญญาณวิทยุ HC-05



รูปที่ 5.12 ผลการทดสอบการส่งข้อมูลของเครื่องรับส่งสัญญาณวิทยุ Xbee PRO S2

ผลการทดสอบการสื่อสารของเครื่องรับส่งสัญญาณวิทยุพบว่าระยะการส่งข้อมูลได้ถูกต้อง (มากกว่า 80%) นั้นของแต่เครื่องรับส่งสัญญาณวิทยุมีระยะแตกต่างกันออกไป เนื่องประเภทของเสาอากาศที่ติดตั้งใช้งานและกำลังสัญญาณสูงสุดที่ไม่เท่ากัน เครื่องรับส่งสัญญาณวิทยุ Xbee PRO Zigbee S2 มีระยะรับส่งสัญญาณมากที่สุด แต่ใช้พลังงานและมีราคาจำหน่ายมากที่สุด ส่วนเครื่องรับส่งสัญญาณวิทยุ nRFL01+ มีระยะรับส่งสัญญาณประมาณ 55 เมตร แต่มีราคาจำหน่ายถูกที่สุด และเครื่องรับส่งสัญญาณวิทยุ HC-05 ที่ใช้เทคโนโลยีบูลทูทมีระยะรับส่งสัญญาณที่น้อยที่สุดประมาณ 6 เมตร เมื่อเทียบกับเครื่องรับส่งสัญญาณเครื่องอื่น ๆ แต่การรับส่งสัญญาณสามารถใช้งานกับอุปกรณ์มาตรฐาน เช่น คอมพิวเตอร์ โทรศัพท์ ได้ง่ายและสะดวกมากกว่าเครื่องรับส่งสัญญาณประเภทอื่น

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 6
สรุปผลและแนวทางการพัฒนาต่อในอนาคต

6.1 สรุปผลการดำเนินงาน

วิทยานิพนธ์นี้นำเสนอการแนวทางการพัฒนาและทางเลือกสำหรับการใช้งานเครือข่ายเซนเซอร์ไร้สายแบ่งได้เป็น 2 ส่วน ได้แก่

ส่วนที่ 1 ต้นแบบโนดเซนเซอร์พัฒนาขึ้นเพื่อแก้ไขข้อจำกัดการใช้งานของโนดเซนเซอร์ที่มีจำหน่ายและใช้เป็นแนวทางการสร้างโนดเซนเซอร์ขึ้นมาใช้งานในประเทศ พัฒนาอ้างอิงกับแพลตฟอร์มอาดูอิโนที่ได้รับความนิยมและใช้งานง่าย แผงวงจรหลักมีขนาดกว้าง 70 มิลลิเมตร ยาว 125 มิลลิเมตร ไมโครคอนโทรลเลอร์ที่เลือกใช้มีหน่วยความจำและแรมขนาดใหญ่ ผู้ใช้สามารถพัฒนาโปรแกรมประยุกต์ที่มีขนาดใหญ่และซับซ้อนได้ ช่องต่อขยายทั้ง 2 ช่องมีการเชื่อมขาสัญญาณของไมโครคอนโทรเลอร์ออกมาให้ใช้งานอย่างครบถ้วน ติดตั้งวงจรฐานเวลาจริงสำหรับอ้างอิงเวลาและการตั้งปลุก วงจรตรวจวัดแบตเตอรี่เพื่อให้ข้อมูลพลังงานที่เหลือในแบตเตอรี่สำหรับการปรับรูปแบบการทำงานและวงจรเพิ่มแรงดันไฟฟ้าจากแบตเตอรี่ สามารถใช้งานคลังโปแกรมและซอฟต์แวร์การพัฒนาได้เช่นเดียวกับบอร์ดรุ่นมาตรฐาน ทำให้ต้นแบบโนดเซนเซอร์ที่นำเสนอในงานวิจัยนี้มีความเหมาะสมในการนำมาใช้งานทั้งงานวิจัยและใช้งานจริงมากกว่าการเลือกบอร์ดอาดูอิโนมาตรฐานมาต่อใช้งานเป็นโนดเซนเซอร์ เนื่องจากยังขาดวงจรที่จำเป็นต่อการใช้งานข้างต้น การต่อเพิ่มวงจรภายนอกเองจะมีความยุ่งยากและเกิดข้อผิดพลาดได้ง่ายสำหรับผู้ใช้งานที่ไม่มีความรู้ด้านอิเล็กทรอนิกส์ ตลอดจนการสั่งซื้อโนดเซนเซอร์จากต่างประเทศที่ไม่ยืดหยุ่นต่อการใช้งานและราคาจำหน่ายสูงอีกด้วย

ส่วนที่ 2 ส่วนต่อประสานโปรแกรมประยุกต์ Simple-API พัฒนาขึ้นโดยใช้รูปแบบการเขียนโปรแกรมเชิงวัตถุและสร้างเป็นคลังโปรแกรมสำหรับเรียกใช้งานในโปรแกรมประยุกต์ของผู้ใช้ ส่วนต่อประสานโปรแกรมประยุกต์ที่นำเสนอสามารถลดจำนวนฟังก์ชันที่ผู้ใช้งานจำเป็นต้องศึกษาก่อนการใช้งานโดยการซ่อนคำสั่งที่เป็นการเขียนโปรแกรมระดับล่างที่มีความซับซ้อนไว้ การรวมกลุ่มฟังก์ชันที่มีหน้าที่การทำงานคล้ายกันไว้ด้วยกันทำให้ผู้ใช้งานสามารถเรียกใช้งานได้ตามความต้องการ ตลอดจนการป้องกันความผิดพลาดในการเรียกใช้งานในขั้นตอนการพัฒนารหัสต้นฉบับและการนำไปใช้งานจริง นอกจากนี้ส่วนต่อประสานโปรแกรมประยุกต์ยังช่วยให้การพัฒนารหัสต้นฉบับของโปรแกรมประยุกต์ไม่ผูกติดกับการเรียกใช้ฟังก์ชันภายในโปรแกรมขับอุปกรณ์นั้นเหมือนกับการพัฒนาโปรแกรมประยุกต์สำหรับ Waspmote หรืออาดูอิโนที่มีการเรียกใช้ฟังก์ชันโปรแกรมขับอุปกรณ์

โดยตรงในโปรแกรมประยุกต์ หากต้องการปรับเปลี่ยนไปใช้อุปกรณ์ตัวใหม่สามารถแก้ไขเฉพาะคลาสภายใน Simple-API และปรับการอาร์กิวเมนต์ของฟังก์ชันที่เรียกใช้ภายในโปรแกรมประยุกต์เพียงเล็กน้อยเท่านั้น แนวคิดการออกแบบใช้งาน Simple-API ยังมีจุดเด่นในการนำไปประยุกต์ใช้ในงานที่มีความซับซ้อนมากขึ้นโดยสามารถแบ่งกลุ่มนักพัฒนาให้แยกพัฒนาในส่วนโปรแกรมประยุกต์และโปรแกรมขับอุปกรณ์ออกจากกันได้เพื่อให้การพัฒนาในแต่ละส่วนมีประสิทธิภาพมากขึ้นและใช้เวลาลดลงกว่าเดิม

การใช้งานต้นแบบโนดเซนเซอร์ร่วมกับส่วนต่อประสานโปรแกรมประยุกต์ Simple-API จะทำให้การประยุกต์ใช้งานเครือข่ายเซนเซอร์ไร้สายได้สะดวกมากขึ้น ลดค่าใช้จ่ายในการนำเข้าโนดเซนเซอร์จากต่างประเทศสามารถปรับแต่งได้ตามความต้องการ นอกจากนี้ผู้ใช้ยังสามารถพัฒนาโปรแกรมประยุกต์เพื่อสั่งงานได้ง่ายมากขึ้น โปรแกรมประยุกต์ที่พัฒนาแล้วสามารถปรับเปลี่ยนได้ง่ายและไม่ต้องกังวลกับการใช้งานฮาร์ดแวร์และการเขียนโปรแกรมระดับล่างด้วย

6.2 แนวทางพัฒนาต่อในอนาคต

การพัฒนาต่อในอนาคตอาจเพิ่มการเชื่อมต่อกับเครือข่ายคอมพิวเตอร์ไร้สายหรือเครือข่ายอินเทอร์เน็ตของโทรศัพท์เคลื่อนที่และการพัฒนาโปรแกรมประยุกต์สำเร็จรูปติดตั้งไว้ภายในโนดเซนเซอร์เพื่อเป็นทางเลือกในการใช้งานเครือข่ายเซนเซอร์ไร้สายที่ง่ายมากและสะดวกมากยิ่งขึ้น สำหรับผู้ใช้งานทั่วไปสามารถให้ผู้ใช้งานแกะออกจากกล่องกำหนดรายละเอียดของการทำงานรูปแบบการทำงานและตรวจวัด รูปแบบการสื่อสาร ผ่านโปรแกรมประยุกต์บนโทรศัพท์ได้แล้วนำไปใช้งานได้เลยทันทีไม่ต้องพัฒนาโปรแกรมประยุกต์เพิ่มเติมอีกต่อไป

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

รายการอ้างอิง

[1] G. V. Merrett and Y. K. Tan, *Wireless sensor networks: application-centric design*: InTech, 2010.

[2] Kazem Sohraby, Daniel Minoli, and T. Znati, "Introduction and Overview of Wireless Sensor Networks," in *Wireless sensor networks: technology, protocols, and applications*, ed New Jersey: John Wiley & Sons, 2007, pp. 1-37.

[3] Kazem Sohraby, Daniel Minoli, and T. Znati, *Wireless sensor networks: technology, protocols, and applications*. New Jersey USA: John Wiley & Sons, 2007.

[4] A. Sharma, R. Chaki, and U. Bhattacharya, "Applications of wireless sensor network in Intelligent Traffic System: A review," in *Electronics Computer Technology (ICECT),* 2011 3*rd International Conference on*, 2011, pp. 53-57.

[5] V. W. Tang, Y. Zheng, and J. Cao, "An intelligent car park management system based on wireless sensor networks," in *Pervasive Computing and Applications,* 2006 1*st International Symposium on*, 2006, pp. 65-70.

[6] S. Zhang and H. Zhang, "A review of wireless sensor networks and its applications," in *Automation and Logistics (ICAL),* 2012 *IEEE International Conference on*, 2012, pp. 386-389.

[7] E. D. Zubiete, L. F. Luque, A. V. M. Rodríguez, and I. G. González, "Review of wireless sensors networks in health applications," in *Engineering in Medicine and Biology Society, EMBC,* 2011 *Annual International Conference of the IEEE*, 2011, pp. 1789-1793.

[8] L. Ruiz-Garcia, L. Lunadei, P. Barreiro, and I. Robla, "A review of wireless sensor technologies and applications in agriculture and food industry: state of the art and current trends," *sensors,* vol. 9, pp. 4728-4750, 2009.

[9] W. Cheunta, N. Chirdchoo, and K. Saelim, "Efficiency improvement of an integrated giant freshwater-white prawn farming in Thailand using a Wireless Sensor Network," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA),* 2014 *Asia-Pacific*, 2014, pp. 1-5.

[10] S. K. Korkua and K. Thinsurat, "Design of ZigBee based WSN for smart demand responsive home energy management system," in *Communications and Information Technologies (ISCIT),* 2013 13*th International Symposium on*, 2013, pp. 549-554.

[11] W. Naruephiphat, R. Promya, and A. Niruntasukrat, "Remote air conditioning control system based on ZigBee Wireless Sensor Network for building," in 2015 *International Computer Science and Engineering Conference (ICSEC)*, 2015, pp. 1-6.

[12] W. Suntiamorntut, S. Charoenpanyasak, and J. Ruksachum, "An elderly assisted living system with wireless sensor networks," in *Wireless and Mobile Networking Conference (WMNC),* 2011 4*th Joint IFIP*, 2011, pp. 1-6.

[13] C. Thanawattano, R. Pongthornseri, and S. Dumnin, "Wearable wireless ECG sensor with cross-platform real-time monitoring," in *Biomedical Engineering and Sciences (IECBES),* 2012 *IEEE EMBS Conference on*, 2012, pp. 284-287.

[14] J. Arnil, Y. Punsawad, and Y. Wongsawat, "Wireless sensor network-based smart room system for healthcare monitoring," in *Robotics and Biomimetics (ROBIO),* 2011 *IEEE International Conference on*, 2011, pp. 2073-2076.

[15] กระทรวงเทคโนโลยีสารสนเทศและการสื่อสาร, "กรอบนโยบายเทคโนโยยีสารสนเทศและการสื่อสารระยะ พ.ศ. ๒๕๕๔-๒๕๖๓ ของประเทศไทย," กระทรวงเทคโนโลยีสารสนเทศและการสื่อสาร, Ed., ed, 2011.

[16] M. Healy, T. Newe, and E. Lewis, "Wireless Sensor Node hardware: A review," in *Sensors,* 2008 *IEEE*, 2008, pp. 621-624.

[17] *TinyOS Documentation Wiki* [Online]. Available: http://tinyos.stanford.edu/tinyos-wiki/index.php/Main_Page

[18] *Contiki: The Open Source OS for the Internet of Things* [Online]. Available: http://www.contiki-os.org/index.html

[19] I. Mampentzidou, E. Karapistoli, and A. A. Economides, "Basic guidelines for deploying Wireless Sensor Networks in agriculture," in *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT),* 2012 4*th International Congress on*, 2012, pp. 864-869.

[20] 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.

[21] K. A. U. Menon, P. D, and M. V. Ramesh, "Wireless sensor network for river water quality monitoring in India," in *Computing Communication & Networking Technologies (ICCCNT), 2012 Third International Conference on*, 2012, pp. 1-7.

[22] L. J. A, pez, F. Soto, J. Suard, az, P. Sanchez, *et al.*, "Development of a mote for horticulture based on the SDI-12 standard," in *Industrial Electronics, 2009. IECON '09. 35th Annual Conference of IEEE*, 2009, pp. 2630-2635.

[23] D. Riley and M. Younis, "A modular and power-intelligent architecture for wireless sensor nodes," in *Local Computer Networks (LCN), 2012 IEEE 37th Conference on*, 2012, pp. 304-307.

[24] A. Grauballe, G. P. Perrucci, and F. H. P. Fitzek, "Opensensor - An open wireless sensor platform," in *4th International Mobile Multimedia Communications Conference*, 2008.

[25] *PSU-Mote User Manual (REV. 1A ed.)* [Online]. Available: http://saturn.ee.psu.ac.th/~kittikhun/PSU_Mote/PSU-Mote_Manual.pdf

[26] *Mica 2 Documentation* [Online]. Available: http://www.eol.ucar.edu/isf/facilities/isa/internal/CrossBow/DataSheets/mica2.pdf

[27] *Mica Z Documentation* [Online]. Available: http://www.memsic.com/userfiles/files/Datasheets/WSN/micaz_datasheet-t.pdf

[28] *IRIS Mote Documentation* [Online]. Available: http://www.memsic.com/userfiles/files/Datasheets/WSN/IRIS_Datasheet.pdf

[29] *TelosB Documentation* [Online]. Available: http://www.willow.co.uk/TelosB_Datasheet.pdf

[30] *Tmote Sky Documentation* [Online]. Available: http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf

Chulalongkorn University

[31] *WISENSE Documentation* [Online]. Available: http://www.wisense.in/api/html/SystemOverview.html

[32] *PanStamp AVR* [Online]. Available: http://panstamp.org/store/index.php?id_product=8&controller=product

[33] *Waspmote Document* [Online]. Available: http://www.libelium.com/development/waspmote/documentation/waspmote-datasheet

[34] C. A. Bell, "Raspberry Pi-based Sensor Nodes," in *Beginning sensor networks with Arduino and Raspberry Pi*, ed New York: Apress, 2013, pp. 97-144.

[35] *Raspberry Pi Products* [Online]. Available: https://www.raspberrypi.org/downloads

[36] *Raspberi Pi 3 Model B* [Online]. Available: http://uk.rs-online.com/web/generalDisplay.html?id=raspberrypi

[37] *Raspbian Installing operating system image* [Online]. Available: https://www.raspberrypi.org/documentation/installation/installing-images/README.md

[38] *Ubuntu MATE for the Raspberry Pi 2 and Raspberry Pi 3* [Online]. Available: https://ubuntu-mate.org/raspberry-pi

[39] S. Pampling. (2015, September 5). *Welcome to RISC OS Pi* [Online]. Available: https://www.riscosopen.org/wiki/documentation/show/Welcome%20to%20RISC%20OS%20Pi

[40] N. Heath. (2015, August 11). *Windows 10 on the Raspberry Pi: What you need to know* [Online]. Available: http://www.techrepublic.com/article/windows-10-on-the-raspberry-pi-what-you-need-to-know

[41] S. F. Barrett and J. Kridner, *Bad to the Bone: Crafting Electronic Systems with BeagleBone and BeagleBone Black* vol. 8: Morgan & Claypool Publishers, 2013.

[42] D. Molloy, *Exploring BeagleBone Tool and Techniques for Building with Embedded Linux*. Indianapolis, IN: Wiley, 2014.

[43] Jkridner. (2016, May 26). *BeagleBone Black* [Online]. Available: http://beagleboard.org/black

[44] Jkridner. (2016, May 2). *Software Support* [Online]. Available: http://beagleboard.org/Support/Software+Support

[45] *NodeMCU Documentation* [Online]. Available: http://nodemcu.readthedocs.io/en/dev/

[46] P. Kromer. (2015, September 2). *Welcome to the nodemcu-devkit wiki!* [Online]. Available: https://github.com/nodemcu/nodemcu-devkit/wiki

[47] ห้องปฏิบัติการระบบสมองกลฝังตัว, ภาควิชาวิศวกรรมศาสตร์และคอมพิวเตอร์ คณะวิศวกรรมศาสตร์, มหาวิทยาลัยพระจอมเกล้าพระนครเหนือ (KMUTNB). (2015). การใช้งานบอร์ด *NodeMCU v2* [Online]. Available: http://cpre.kmutnb.ac.th/esl/learning/index.php?article=nodemcu_v2

[48] ห้องปฏิบัติการระบบสมองกลฝังตัว, ภาควิชาวิศวกรรมศาสตร์และคอมพิวเตอร์, คณะวิศวกรรมศาสตร์, มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ (KMUTNB). (2015). การใช้งาน *NodeMCU* สำหรับโมดูล *ESP*8266 เบื้องต้น [Online]. Available: http://cpre.kmutnb.ac.th/esl/learning/index.php?article=esp8266-nodemcu

[49] J. Purdum, *Beginning C for Arduino: Learn C Programming for the Arduino and Compatible Microcontrollers*: Apress, 2012.

[50] *Arduino Products* [Online]. Available: https://www.arduino.cc/en/Main/Products

[51] J. Bayle, "First Contact with C," in *C Programming for Arduino*, ed Birmingham, UK: Packt Publishing, 2013.

[52] *The Arduino Playground* [Online]. Available: http://playground.arduino.cc

[53] maniacbug. (2011). *Arduino on ATmega1284P* [Online]. Available: https://maniacbug.wordpress.com/2011/11/27/arduino-on-atmega1284p-4/

ภาคผนวก

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

การใช้งานพารามิเตอร์ของคลาส ProcessingSystemC

P_MOTE_NAME สำหรับเก็บชื่อของโนดเซนเซอร์สามารถอ่านและเขียนได้ ข้อมูลจะเก็บอยู่ในรหัสแอสกีขนาดเท่ากับ 50 ไบต์

ลำดับ	1	2	3	4	...	49	50
P_MOTE_NAME					...		
ค่า	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII

P_GROUP_ID สำหรับเก็บหมายเลขประจำกลุ่มของโนดเซนเซอร์ สามารถอ่านและเขียนได้ ขนาดเท่ากับ 10 ไบต์

ลำดับ	1	2	3	4	...	9	10
P_GROUP_ID					...		
ค่า	0-255	0-255	0-255	0-255	0-255	0-255	0-255

P_MOTE_ADDRESS สำหรับเก็บกลุ่มหมายเลขประจำตัวของโนดเซนเซอร์ สามารถอ่านและเขียนได้ ความยาว 8 ไบต์

ลำดับ	1	2	3	4	...	7	8
P_MOTE_ADDRESS					...		
ค่า	0-255	0-255	0-255	0-255	0-255	0-255	0-255

P_RADIO_ADDRESS สำหรับเก็บกลุ่มหมายเลขประจำตัวของเครื่องรับส่งสัญญาณวิทยุที่ติดตั้งบนโนดเซนเซอร์ สามารถอ่านและเขียนได้ ความยาว 10 ไบต์

ลำดับ	1	2	3	4	...	9	10
P_RADIO_ADDRESS					...		
ค่า	0-255	0-255	0-255	0-255	0-255	0-255	0-255

P_ON_BOARD_TEMP สำหรับเก็บข้อมูลอุณหภูมิบนแผงวงจร ตรวจวัดได้จากเซนเซอร์ภายในวงจรฐานเวลาจริง สามารถอ่านค่าได้เพียงอย่างเดียว ความยาว 1 ไบต์

ลำดับ	1
P_ON_BOARD_TEMP	temp
ค่า	0-85

temp	ข้อมูลอุณหภูมิที่อ่านได้บนแผงวงจร (หน่วยองศาเซลเซียส)

P_SLEEP_MODE สำหรับกำหนดโหมดการทำงานพลังงานต่ำของไมโครคอนโทรลเลอร์ สามารถเขียนค่าได้อย่างเดียว ค่าเริ่มต้นจะใช้โหมด SLEEP_MODE_PWR_DOWN ความยาว 1 ไบต์

ลำดับ	1
P_SLEEP_MODE	mode
ค่า	1-2

mode	หมายเลขโหมดการทำงานพลังงานต่ำ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ชื่อค่าคงที่	หมายเลข	คำอธิบาย
SLEEP_MODE_IDLE	0	โหมดการทำงานที่ใช้พลังงานต่ำโดยหยุดการทำงานของหน่วยประมวลผลแต่ช่องต่อประสานอื่น ๆ ยังคงสามารถทำงาน
SLEEP_MODE_PWR_DOWN	3	โหมดการทำงานที่ใช้พลังงานต่ำสุดโดยหยุดการทำงานของหน่วยประมวลผลและสื่อสารผ่านช่องต่อประสานอื่น ๆ

P_CLK_DIV สำหรับกำหนดตัวหารความถี่สัญญาณนาฬิกาภายในไมโครคอนโทรลเลอร์ ค่าเริ่มต้นจะถูกกำหนดให้ตัวหารความถี่มีค่าเท่ากับ 1 เมื่อผู้ใช้เปลี่ยนตัวหารความถี่จะส่งผลกระทบต่อการสื่อสารและการใช้พลังงานของไมโครคอนโทรลเลอร์ สามารถเขียนค่าได้เพียงอย่างเดียว ความยาว 1 ไบต์

ลำดับ	1
P_CLK_DIV	div_clk
ค่า	0-2

div_clock	หมายเลขตัวหารความถี่ภายในของไมโครคอนโทรลเลอร์

ชื่อค่าคงที่	หมายเลข	คำอธิบาย
clock_div_1	0	ตัวหารเท่ากับ 1 (ความถี่ 16 เมกะเฮิรตซ์)
clock_div_2	1	ตัวหารเท่ากับ 2 (ความถี่ 8 เมกะเฮิรตซ์)
clock_div_4	2	ตัวหารเท่ากับ 4 (ความถี่ 4 เมกะเฮิรตซ์)

P_TIME สำหรับเก็บและกำหนดเวลาจากวงจรฐานเวลาจริง เวลาจะเก็บอยู่ในรูปแบบ 24 ชั่วโมง สามารถอ่านและเขียนค่าได้ มีความยาว 3 ไบต์

ลำดับ	1	2	3
P_TIME	hrs	min	sec
ค่า	0-23	0-59	0-59

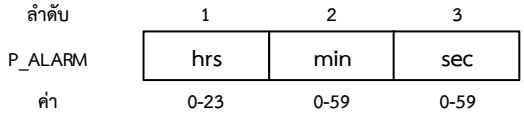
hrs	ข้อมูลชั่วโมง (รูปแบบ 0 – 23)
min	ข้อมูลนาที (รูปแบบ 0 – 59)
sec	ข้อมูลวินาที (รูปแบบ 0 – 59)

P_DATE สำหรับเก็บและกำหนดวันที่จากวงจรฐานเวลาจริง สามารถอ่านและเขียนค่าได้ มีความยาว 4 ไบต์

ลำดับ	1	2	3	4
P_DATE	day	date	month	year
ค่า	1-7	1-31	1-12	0-99

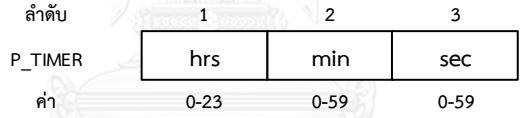
day	ข้อมูลวันในสัปดาห์ (1=อาทิตย์ 2=จันทร์ 3=อังคาร 4=พุธ 5=พฤหัสบดี 6=ศุกร์ 7=เสาร์)
date	ข้อมูลวันที่ (รูปแบบ 1 – 31)
month	ข้อมูลเดือน (รูปแบบ 1 – 12)
year	ข้อมูลปี ค.ศ. (รูปแบบ 2 หลัก)

P_ALARM สำหรับเก็บเวลาตั้งปลุกของวงจรฐานเวลาจริงเพื่อปลุกไมโครคอนโทรลเลอร์จากการทำงานโหมดนอนหลับ โดยข้อมูลเวลาปลุกจะเก็บอยู่ในรูปแบบ 24 ชั่วโมง คล้ายกับการตั้งนาฬิกาปลุกทั่วไป สามารถเขียนค่าได้เพียงอย่างเดียว มีความยาว 3 ไบต์



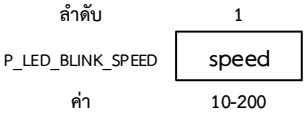
hrs	ข้อมูลชั่วโมง (รูปแบบ 0 – 23)
min	ข้อมูลนาที (รูปแบบ 0 – 59)
sec	ข้อมูลวินาที (รูปแบบ 0 – 59)

P_TIMER สำหรับเก็บเวลาตั้งปลุกของวงจรฐานเวลาจริงเพื่อปลุกไมโครคอนโทรลเลอร์จากการทำงานโหมดนอนหลับ โดยข้อมูลระยะเวลาปลุกจะเก็บอยู่ในรูปแบบ ชั่วโมง นาที และ วินาที ทำงานแบบนาฬิกานับเวลาถอยหลัง สามารถเขียนค่าได้เพียงอย่างเดียว มีความยาว 3 ไบต์



hrs	ข้อมูลชั่วโมง (รูปแบบ 0 – 23)
min	ข้อมูลนาที (รูปแบบ 0 – 59)
sec	ข้อมูลวินาที (รูปแบบ 0 – 59)

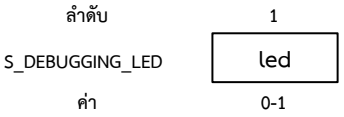
P_LED_BLINK_SPEED พารามิเตอร์สำหรับเก็บระยะเวลาในการเปิดและปิดหลอดแอลอีดี สามารถเขียนค่าได้เพียงอย่างเดียว มีความยาว 1 ไบต์



speed	ระยะเวลา (หน่วยมิลลิวินาที)

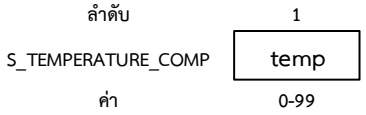
การใช้งานพารามิเตอร์ของคลาส SensorAndActuatorC

S_DEBUGGING_LED พารามิเตอร์สำหรับสั่งเปิดหรือปิดการใช้งานหลอดแอลอีดีเพื่อตรวจสอบการทำงานบนวงจรเซนเซอร์ สามารถเขียนค่าได้เพียงอย่างเดียว ขนาดเท่ากับ 1 ไบต์



led	ค่า 0 – ปิดใช้งานหลอดแอลอีดีสำหรับตรวจสอบการทำงาน ค่า 1 – เปิดใช้งานหลอดแอลอีดีสำหรับตรวจสอบการทำงาน

S_TEMPERATURE_COMP พารามิเตอร์สำหรับกำหนดค่าอุณหภูมิของสารละลาย เพื่อเป็นข้อมูลในการคำนวณหาค่า D.O. (ค่าอุณหภูมิตั้งต้นเท่ากับ 25 องศาเซลเซียส) สามารถเขียนค่าได้เพียงอย่างเดียว ขนาดเท่ากับ 4 ไบต์



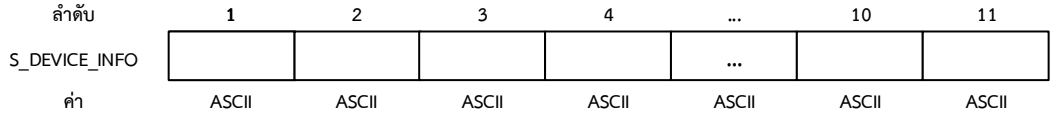
temp	ข้อมูลอุณหภูมิเพื่อใช่คำนวณหาค่าพีเอช (หน่วยองศาเซลเซียส)

S_CONDUCTIVITY_COMP พารามิเตอร์สำหรับกำหนดค่าการนำไฟฟ้า (conductivity) ของน้ำที่ทำการวัดเพื่อเป็นข้อมูลในการคำนวณหาค่า D.O. (ค่าตั้งต้นเท่ากับ 0) ผู้ใช้สามารถกำหนดค่าการนำไฟฟ้าได้ด้วยตัวเองหรือใช้ค่ามาตรฐานดังแสดงใน Error! Reference source not found. สามารถเขียนค่าได้เพียงอย่างเดียว ขนาดเท่ากับ 4 ไบต์

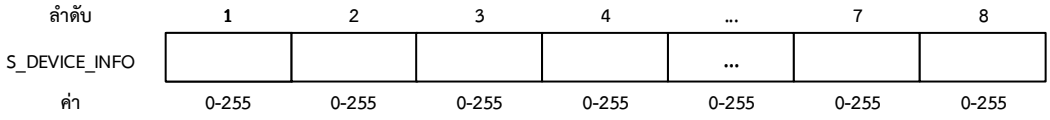
cond	ค่าการนำไฟฟ้าในน้ำ (หน่วย µs)

ชื่อค่าคงที่	ค่าการนำไฟฟ้า (µs)	คำอธิบาย
CONDUCT_FRESH_WATER	0	ค่าการนำไฟฟ้าในน้ำจืด
CONDUCT_BRACKISH_WATER	19,900	ค่าการนำไฟฟ้าในน้ำกร่อย
CONDUCT_SALT_WATER	54,000	ค่าการนำไฟฟ้าในน้ำเค็ม

S_DEVICE_ INFO พารามิเตอร์สำหรับเก็บข้อมูลหมายเลขรุ่นและวันที่ของเฟิร์มแวร์ภายในวงจรเซนเซอร์ ข้อมูลจะเก็บอยู่ในรหัสแอสกี สามารถอ่านค่าได้เพียงอย่างเดียว ขนาดเท่ากับ 11 ไบต์

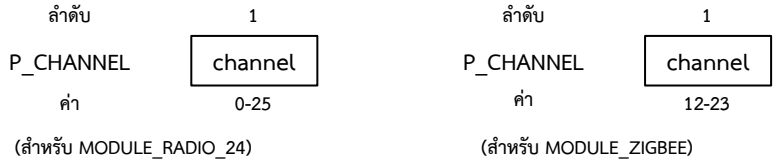


S_DEVICE_INFO พารามิเตอร์สำหรับเก็บข้อมูลหมายเลขเซนเซอร์ ข้อมูลจะเก็บอยู่ในรูปตัวเลข สามารถอ่านค่าได้เพียงอย่างเดียว ขนาดเท่ากับ 8 ไบต์ *(สำหรับมอดูล MODULE_DALLAS_1820)*



การใช้งานพารามิเตอร์ของคลาส WirelessCommunicationSystemC

P_CHANNEL พารามิเตอร์สำหรับกำหนดช่องสัญญาณในการสื่อสารไร้สาย เครื่องส่งสัญญาณวิทยุที่จะต้องใช้ช่องสัญญาณเดียวกันในการสื่อสารระหว่างโนด สามารถเขียนได้อย่างเดียว ขนาดเท่ากับ 1 ไบต์



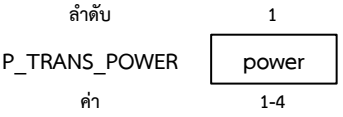
channel	หมายเลขช่องสัญญาณที่ใช้สื่อสาร

P_DATA_RATE พารามิเตอร์สำหรับกำหนดความเร็วในการสื่อสารไร้สาย เครื่องส่งสัญญาณวิทยุที่จะต้องใช้ช่องสัญญาณเดียวกันในการสื่อสารระหว่างโนด สามารถเขียนได้อย่างเดียว ขนาดเท่ากับ 1 ไบต์

rate	แรงดันไฟฟ้าของแบตเตอรี่ขาเข้าวงจรตรวจวัด (หน่วยโวลต์)

ชื่อค่าคงที่	ค่า	อัตราเร็ว
NRF24_DATARATE_1M	1	1 Mbps
NRF24_DATARATE_2M	2	2 Mbps
NRF24_DATARATE_250K	0	250 kbps

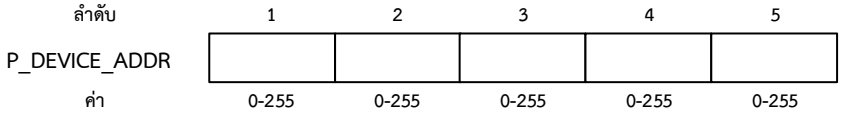
P_TRANS_POWER พารามิเตอร์สำหรับกำหนดกำลังที่ใช้ส่งสัญญาณวิทยุ เครื่องส่งสัญญาณวิทยุที่จะต้องใช้ช่องสัญญาณเดียวกันในการสื่อสารระหว่างโนด สามารถเขียนได้อย่างเดียว ขนาดเท่ากับ 1 ไบต์



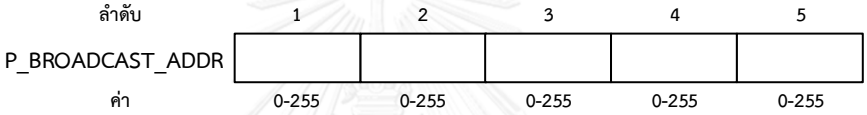
power	กำลังในการส่งสัญญาณ

ชื่อค่าคงที่	ค่า	กำลัง
NRF24_POWER_TRANS_25	1	-18 dBm
NRF24_POWER_TRANS_50	2	-12 dBm
NRF24_POWER_TRANS_75	3	-6 dBm
NRF24_POWER_TRANS_100	4	0 dBm

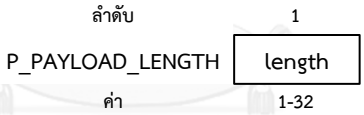
P_DEVICE_ADDR พารามิเตอร์สำหรับกำหนดหมายเลขประจำเครื่องส่งสัญญาณวิทยุ ใช้สำหรับกรองข้อมูลที่รับมา สามารถเขียนได้เพียงอย่างเดียว ขนาดเท่ากับ 5 ไบต์



P_BROADCAST_ADDR พารามิเตอร์สำหรับกำหนดหมายเลขสำหรับกรองข้อความบรอดคาสต์ (broadcast) ภายในเครือข่าย สามารถเขียนได้เพียงอย่างเดียว ขนาดเท่ากับ 5 ไบต์

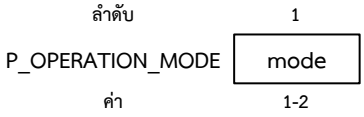


P_PAYLOAD_LENGTH พารามิเตอร์สำหรับกำหนดความยาวของข้อมูลที่ใช้ส่ง เครื่องส่งสัญญาณวิทยุที่จะต้องมีการกำหนดความยาวของข้อมูลขนาดเดียวกันในการสื่อสารระหว่างโนด สามารถเขียนได้อย่างเดียว ขนาดเท่ากับ 1 ไบต์



length	ขนาดความยาวข้อมูล (ไบต์)

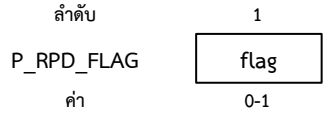
P_OPERATION_MODE พารามิเตอร์สำหรับกำหนดโหมดการทำงานของเครื่องส่งสัญญาณวิทยุ สามารถอ่านและเขียนได้ตลอดการทำงาน ขนาดเท่ากับ 1 ไบต์



mode	โหมดการทำงานของเครื่องรับส่งสัญญาณวิทยุ

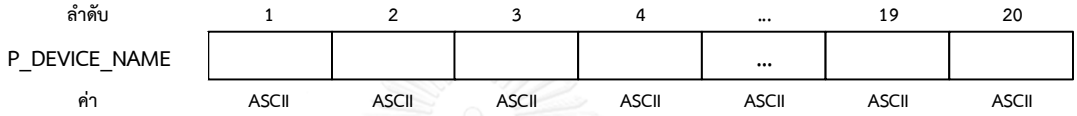
ชื่อค่าคงที่	ค่า	คำอธิบาย
STATE_POWER_DOWN	0	หยุดการทำงานทั้งหมด (ต้องตั้งค่าใหม่ทั้งหมด)
STATE_STANDBY	1	หยุดการทำงานชั่วคราวในภาครับและส่ง
STATE_RX_MODE	2	รอรับข้อมูล

P_RDP_FLAG พารามิเตอร์สำหรับแสดงค่าพลังงานของช่องสัญญาณ (received power detector – RDP) สามารถอ่านค่าได้เพียงอย่างเดียวขณะเครื่องรับส่งสัญญาณวิทยุทำงานในโหมดรับข้อมูลเท่านั้น ขนาดเท่ากับ 1 ไบต์

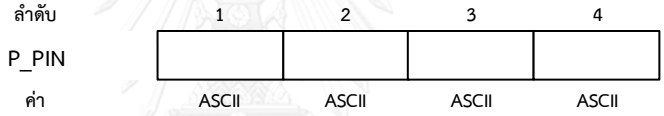


flag	1 สัญญาณพาห์มีกำลังมากกว่า -64 dBm 0 สัญญาณพาห์มีกำลังน้อยกว่า -64 dBm

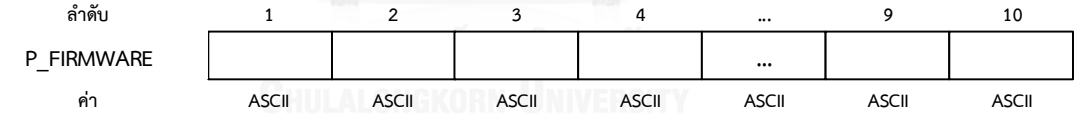
P_DEVICE_NAME พารามิเตอร์สำหรับกำหนดชื่อเครื่องรับส่งสัญญาณวิทยุ โดยจะใช้เป็นชื่อในขั้นตอนการจับคู่อุปกรณ์ สามารถเขียนค่าได้เพียงอย่างเดียวและทำขณะเครื่องรับส่งสัญญาณยังไม่ได้จับคู่อุปกรณ์ ขนาดเท่ากับ 20 ไบต์



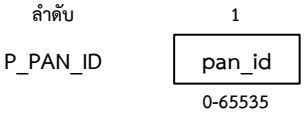
*P*_PIN พารามิเตอร์สำหรับกำหนดรหัสผ่าน 4 ตัวสำหรับการจับคู่อุปกรณ์ สามารถเขียนค่าได้เพียงอย่างเดียวและทำขณะเครื่องรับส่งสัญญาณยังไม่ได้จับคู่อุปกรณ์ ขนาดเท่ากับ 4 ไบต์



P_FIRMWARE พารามิเตอร์สำหรับอ่านค่าเฟิร์มแวร์ภายในเครื่องรับส่งสัญญาณวิทยุ สามารถอ่านค่าได้เพียงอย่างเดียวและทำขณะเครื่องรับส่งสัญญาณยังไม่ได้จับคู่อุปกรณ์ ขนาดเท่ากับ 10 ไบต์

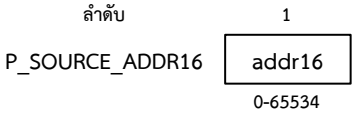


P_PAN_ID พารามิเตอร์สำหรับตั้งค่าหมายเลขเครือข่ายส่วนบุคคล (PAN) สามารถเขียนค่าได้เพียงอย่างเดียวและทำขณะก่อนการสื่อสารข้อมูล ขนาดเท่ากับ 2 ไบต์



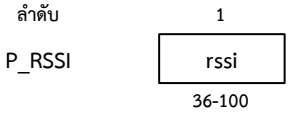
pan_id	หมายเลขประจำเครือข่าย (หมายเลข 65535 สงวนไว้สำหรับการส่งข้อความบรอดคาสต์ภายในเครือข่าย)

P_SOURCE_ADD16 พารามิเตอร์สำหรับตั้งค่าหมายเลขประจำตัวของเครื่องรับส่งสัญญาณวิทยุ Xbee สามารถอ่านและเขียนค่าได้ ขนาดเท่ากับ 2 ไบต์



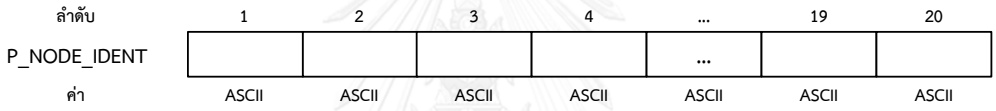
addr16	หมายเลขประจำตัวเครื่องรับส่งสัญญาณวิทยุ

P_RSSI พารามิเตอร์สำหรับอ่านค่าความเข้มของสัญญาณตัวรับ (received signal strength indicator – RSSI) มีหน่วยวัดเป็น dBm สามารถอ่านค่าได้เพียงอย่างเดียว ขนาด 1 ไบต์



rssi	ค่าความเข้มสัญญาณของข้อมูลที่รับตัวปัจจุบัน

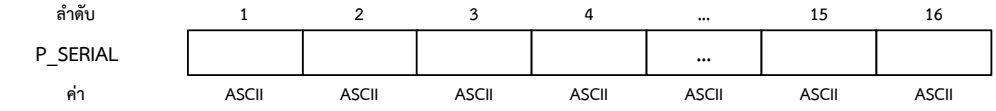
P_NODE_IDENT พารามิเตอร์สำหรับกำหนดชื่อของเครื่องรับส่งสัญญาณวิทยุ Xbee สามารถอ่านและเขียนได้ ขนาดเท่ากับ 20 ไบต์



P_HW_INFO พารามิเตอร์สำหรับเก็บหมายเลขรุ่นของอุปกรณ์ สามารถอ่านได้อย่างเดียว ขนาดเท่ากับ 2 ไบต์

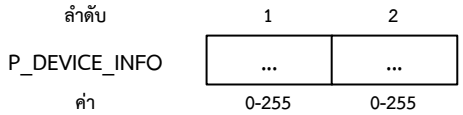
hw_info	หมายเลขรุ่นที่ผลิตเครื่องรับส่งสัญญาณวิทยุ

P_SERIAL พารามิเตอร์สำหรับเก็บหมายเลขประจำตัวเครื่องรับส่งสัญญาณที่ผู้ผลิตกำหนดมาไม่อนุญาตให้แก้ไข สามารถอ่านได้อย่างเดียว ขนาดเท่ากับ 16 ไบต์

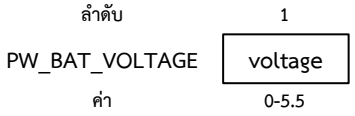


การใช้งานพารามิเตอร์ของคลาส PowerControlSystemC

PW_DEVICE_INFO สำหรับเก็บข้อมูลหมายเลขรุ่นของวงจรรวมตรวจสอบแบตเตอรี่ สามารถอ่านค่าได้เพียงอย่างเดียว ขนาด 1 ไบต์

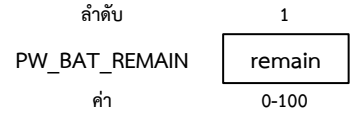


PW_BAT_VOL สำหรับเก็บข้อมูลแรงดันแบตเตอรี่ขาเข้าวงจรตรวจสอบแบตเตอรี่ สามารถอ่านค่าได้เพียงอย่างเดียว ขนาด 4 ไบต์



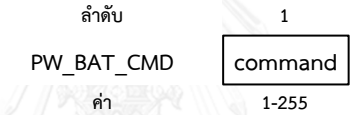
voltage	แรงดันไฟฟ้าของแบตเตอรี่ขาเข้าวงจรตรวจวัด (หน่วยโวลต์)

PW_BAT_REMAIN สำหรับเก็บข้อมูลประจุที่เหลืออยู่ของแบตเตอรี่ที่เหลืออยู่ สามารถอ่านได้เพียงอย่างเดียว ขนาด 4 ไบต์



remain	ปริมาณประจุไฟฟ้าที่เหลืออยู่ในแบตเตอรี่ (หน่วยเปอร์เซ็นต์)

PW_BAT_CMD สำหรับส่งหมายเลขคำสั่งพิเศษไปยังวงจรแบตเตอรี่โดยตรง วงจรแบตเตอรี่จะรับคำสั่งไปทำงานต่อทันที สามารถเขียนค่าได้เพียงอย่างเดียว ขนาด 1 ไบต์



command	หมายเลขคำสั่งพิเศษ

ชื่อค่าคงที่	หมายเลขคำสั่ง	คำอธิบาย
CMD_RESET	1	คำสั่งรีเซตการตรวจวัดแบตเตอรี่

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ฟังก์ชันสำหรับการเข้าถึงพารามิเตอร์และสั่งงาน

1. คลาส ProcessingSystemC

ฟังก์ชันเริ่มการทำงาน

void begin();

ฟังก์ชันสำหรับเริ่มการทำงานการเชื่อมต่อและกำหนดค่าเริ่มต้นของอุปกรณ์ ฟังก์ชันนี้ไม่มีการคืนค่ากลับและต้องถูกเรียกใช้งานเป็นลำดับแรกเสมอ

ฟังก์ชันจัดการพารามิเตอร์

int8_t writeParameter(uint8_t module, uint8_t paramName, uint8_t *value, uint8_t valueLength);

ฟังก์ชันสำหรับกำหนดค่าพารามิเตอร์ของมอดูล สามารถเรียกใช้งานได้ตลอดทั้งโปรแกรมเมื่อต้องการเปลี่ยนแปลงค่าพารามิเตอร์ของมอดูลนั้น ฟังก์ชันนี้มีการคืนค่ากลับเป็นหมายเลขแทนข้อผิดพลาด การเรียกใช้มีอาร์กิวเมนต์ดังต่อไปนี้

module	มอดูลที่ต้องการใช้งาน
paramName	พารามิเตอร์ที่ต้องการเขียนค่า

**value*	พอยเตอร์ของข้อมูลที่ต้องการกำหนดค่าให้พารามิเตอร์
valueLength	ขนาดของข้อมูล (ไบต์)

int8_t readParameter(uint8_t module, uint8_t paramName, uint8_t *value, uint8_t valueLength);

ฟังก์ชันสำหรับอ่านค่าพารามิเตอร์ของมอดูล สามารถเรียกใช้งานได้ตลอดทั้งโปรแกรมเมื่อต้องการอ่านค่าพารามิเตอร์ของมอดูลนั้น ฟังก์ชันนี้มีการคืนค่ากลับเป็นหมายเลขแทนข้อผิดพลาด การเรียกใช้มีอาร์กิวเมนต์ดังต่อไปนี้

module	มอดูลที่ต้องการใช้งาน
paramName	พารามิเตอร์ที่ต้องการอ่านค่า
**value*	พอยเตอร์ของข้อมูลที่ต้องการเก็บค่าจากพารามิเตอร์
valueLength	ขนาดความยาวของข้อมูล (ไบต์)

ฟังก์ชันจัดการข้อมูลผู้ใช้

int8_t writeUserData(uint8_t page, const uint8_t *value, uint8_t length);

ฟังก์ชันสำหรับการเขียนข้อมูลผู้ใช้ลงในหน่วยความจำประเภท EEPROM โดยแบ่งการเขียนเป็นบล็อกที่มีความกว้างขนาด 10 ไบต์ จำนวน 350 ช่อง สามารถเรียกใช้งานได้ตลอดทั้งโปรแกรม ฟังก์ชันนี้มีการคืนค่ากลับเป็นหมายเลขแทนข้อผิดพลาด การเรียกใช้มีอาร์กิวเมนต์ ดังต่อไปนี้

page	หมายเลขบล็อกที่ใช้เก็บข้อมูล (1-350)
**value*	พอยเตอร์ของข้อมูลที่ต้องการเก็บลงหน่วยความจำ
valueLength	ขนาดของข้อมูล (ความยาวสูงสุดไม่เกิน 10 ไบต์)

int8_t readUserData(uint8_t page, uint8_t *value, uint8_t length);

ฟังก์ชันสำหรับการอ่านข้อมูลผู้ใช้จากในหน่วยความจำประเภท EEPROM โดยแบ่งการเขียนเป็นบล็อก (block) ขนาด 10 ไบต์ จำนวน 350 ช่อง สามารถเรียกใช้งานได้ตลอดทั้งโปรแกรม ฟังก์ชันนี้มีการคืนค่ากลับเป็นหมายเลขแทนข้อผิดพลาด การเรียกใช้มีอาร์กิวเมนต์ ดังต่อไปนี้

page	หมายเลขบล็อกที่ใช้เก็บข้อมูล (1-350)
**value*	พอยเตอร์ของข้อมูลที่อ่านจากหน่วยความจำ
valueLength	ขนาดของข้อมูล (ความยาวสูงสุดไม่เกิน 10 ไบต์)

ฟังก์ชันควบคุมหลอดแอลอีดีแสดงสถานะ

void onBoardLedOn();

ฟังก์ชันสำหรับเปิดใช้งานหลอดแอลอีดีบนแผงวงจรหลัก สามารถเรียกใช้งานได้ตลอดทั้งโปรแกรม ฟังก์ชันนี้ไม่มีการคืนค่ากลับและไม่มีอาร์กิวเมนต์

void onBoardLedOff();

ฟังก์ชันสำหรับปิดใช้งานหลอดแอลอีดีบนแผงวงจรหลัก สามารถเรียกใช้งานได้ตลอดทั้งโปรแกรม ฟังก์ชันนี้ไม่มีการคืนค่ากลับและไม่มีอาร์กิวเมนต์

void onBoardLedBlink(uint8_t times);

ฟังก์ชันสำหรับปิดใช้งานหลอดแอลอีดีบนแผงวงจรหลัก สามารถเรียกใช้งานได้ตลอดทั้งโปรแกรม ฟังก์ชันนี้ไม่มีการคืนค่ากลับ โดยมีอาร์กิวเมนต์ดังนี้

times	จำนวนครั้งที่ต้องการให้หลอดแอลอีดีกระพริบไฟ (0 หมายถึง กระพริบต่อเนื่อง)

ฟังก์ชันจัดการโหมดการทำงานและรีเซตไมโครคอนโทรลเลอร์

void sleepNow();

ฟังก์ชันสำหรับสั่งเข้าสู่การทำงานในโหมดนอนหลับ ผู้ใช้จะต้องมีการกำหนดค่าพารามิเตอร์ P_ALARM หรือ P_TIMER ของมอดูล MODULE_REAL_TIME ก่อนเรียกใช้คำสั่งนี้ ฟังก์ชันนี้ไม่มีการคืนค่ากลับและไม่มีอาร์กิวเมนต์ เมื่อเรียกใช้งานไมโครคอนโทรเลอร์จะเข้าสู่การทำงานที่ใช้พลังงานต่ำและเมื่อออกจากการทำงานโปรแกรมจะเริ่มทำงานในคำสั่งถัดไปจากคำสั่งนี้

void resetNow();

ฟังก์ชันสำหรับรีเซตการทำงานของไมโครคอนโทรลเลอร์ เมื่อเรียกใช้คำสั่งจะทำการรอเป็นเวลา 100 มิลลิวินาทีก่อนสั่งรีเซตไปยังไมโครคอนโทรลเลอร์ ฟังก์ชันนี้ไม่มีการคืนค่ากลับและไม่มีอาร์กิวเมนต์

2. คลาส SensorAndActuatorC

ฟังก์ชันเริ่มการทำงาน

int8_t begin(uint8_t module, uint8_t interfaceName);

ฟังก์ชันสำหรับเริ่มการเชื่อมต่อและกำหนดค่าเริ่มต้นของอุปกรณ์ คำสั่งภายในจะสร้างการเชื่อมต่อไปยังอุปกรณ์จริงตามชื่อของมอดูลและช่องเชื่อมต่อ ฟังก์ชันนี้มีการคืนค่ากลับเป็นตัวเลขแสดงผลการทำงานและจำเป็นต้องเรียกใช้งานเป็นลำดับแรกในขั้นตอนการใช้งานอุปกรณ์นั้นเสมอ

module	มอดูลที่ต้องการใช้งาน
interfaceName	ชื่อช่องต่อที่อุปกรณ์นั้นใช้งาน

ฟังก์ชันจัดการพารามิเตอร์

int8_t writeParameter(uint8_t module, uint8_t paramName, float *value, uint8_t valueLength)

int8_t writeParameter(uint8_t module, uint8_t paramName, uint8_t *value, uint8_t valueLength);

ฟังก์ชันสำหรับกำหนดค่าพารามิเตอร์ของมอดูล สามารถเรียกใช้งานได้ตลอดทั้งโปรแกรมเมื่อต้องการเปลี่ยนแปลงค่าพารามิเตอร์ของมอดูลนั้น ฟังก์ชันนี้มีการคืนค่ากลับเป็นหมายเลขแทนข้อผิดพลาด การเรียกใช้มีอาร์กิวเมนต์ดังต่อไปนี้

module	มอดูลที่ต้องการใช้งาน
paramName	พารามิเตอร์ที่ต้องการเขียนค่า
**value*	พอยเตอร์ของข้อมูล
valueLength	ขนาดของข้อมูล

int8_t readParameter(uint8_t module, uint8_t paramName, uint8_t *value, uint8_t valueLength);

ฟังก์ชันสำหรับอ่านค่าพารามิเตอร์ของมอดูล สามารถเรียกใช้งานได้ตลอดทั้งโปรแกรมเมื่อต้องการอ่านค่าพารามิเตอร์ของมอดูลนั้น ฟังก์ชันนี้มีการคืนค่ากลับเป็นตัวเลขแทนผลของการทำงาน การเรียกใช้มีอาร์กิวเมนต์ที่เกี่ยวข้องดังต่อไปนี้

module	มอดูลที่ต้องการใช้งาน
paramName	พารามิเตอร์ที่ต้องการอ่านค่า
**value*	พอยเตอร์ของข้อมูล
valueLength	ขนาดของข้อมูล

ฟังก์ชันสั่งงานเซนเซอร์

int8_t calibrateSensor(Stream *ptrDisplay, uint8_t module);

ฟังก์ชันสำหรับปรับตั้งค่ามาตรฐานของเซนเซอร์ สำหรับเซนเซอร์ที่อนุญาตให้ผู้ใช้ปรับตั้งค่าได้เอง การเรียกใช้มีอาร์กิวเมนต์ที่เกี่ยวข้องดังต่อไปนี้

**ptrDisplay*	พอยเตอร์ของการสื่อสารผ่านพอร์ตอนุกรมเพื่อขั้นตอนการทำงาน
module	มอดูลที่ต้องการใช้งาน

int8_t readSensorData(uint8_t module, float *value, uint8_t valueLength);

ฟังก์ชันสำหรับอ่านค่าที่ได้จากการตรวจวัดของเซนเซอร์ สามารถเรียกใช้งานได้หลังจากเรียกใช้ฟังก์ชัน *begin();* การเรียกใช้มีอาร์กิวเมนต์ที่เกี่ยวข้องดังต่อไปนี้

module	มอดูลที่ต้องการใช้งาน
**value*	พอยเตอร์ของข้อมูล
valueLength	ขนาดของข้อมูล

3. คลาส WirelessCommunicationSystemC

ฟังก์ชันเริ่มการทำงาน

int8_t begin(uint8_t module, uint8_t interfaceName);

ฟังก์ชันสำหรับเริ่มการเชื่อมต่อและกำหนดค่าเริ่มต้นของอุปกรณ์ คำสั่งภายในจะสร้างการเชื่อมต่อไปยังอุปกรณ์จริงตามชื่อของมอดูลและช่องเชื่อมต่อ ฟังก์ชันนี้มีการคืนค่ากลับเป็นตัวเลขแสดงผลการทำงานและจำเป็นต้องเรียกใช้งานเป็นลำดับแรกในขั้นตอนการใช้งานอุปกรณ์นั้นเสมอ

module	มอดูลที่ต้องการใช้งาน
interfaceName	ชื่อช่องต่อที่อุปกรณ์นั้นใช้งาน

ฟังก์ชันจัดการพารามิเตอร์

int8_t writeParameter(uint8_t module, uint8_t paramName, float *value,
uint8_t valueLength)
int8_t writeParameter(uint8_t module, uint8_t paramName, uint8_t *value,
uint8_t valueLength);

ฟังก์ชันสำหรับกำหนดค่าพารามิเตอร์ของมอดูล สามารถเรียกใช้งานได้ตลอดทั้งโปรแกรมเมื่อต้องการเปลี่ยนแปลงค่าพารามิเตอร์ของมอดูลนั้น ฟังก์ชันนี้มีการคืนค่ากลับเป็นหมายเลขแทนข้อผิดพลาด การเรียกใช้มีอาร์กิวเมนต์ดังต่อไปนี้

module	มอดูลที่ต้องการใช้งาน
paramName	พารามิเตอร์ที่ต้องการเขียนค่า
**value*	พอยเตอร์ของข้อมูล
valueLength	ขนาดของข้อมูล

int8_t readParameter(uint8_t module, uint8_t paramName, uint8_t *value,
uint8_t valueLength);
int8_t readParameter(uint8_t module, uint8_t paramName, uint8_t *value,
uint8_t valueLength);

ฟังก์ชันสำหรับอ่านค่าพารามิเตอร์ของมอดูล สามารถเรียกใช้งานได้ตลอดทั้งโปรแกรมเมื่อต้องการอ่านค่าพารามิเตอร์ของมอดูลนั้น ฟังก์ชันนี้มีการคืนค่ากลับเป็นตัวเลขแทนผลของการทำงาน การเรียกใช้มีอาร์กิวเมนต์ที่เกี่ยวข้องดังต่อไปนี้

module	มอดูลที่ต้องการใช้งาน
paramName	พารามิเตอร์ที่ต้องการอ่านค่า
**value*	พอยเตอร์ของข้อมูล
valueLength	ขนาดของข้อมูล

ฟังก์ชันสั่งงาน

int8_t sendData(uint8_t module, uint8_t *destination, uint8_t *msg, uint8_t msgLength);

ฟังก์ชันสำหรับส่งข้อมูลผ่านเครื่องรับส่งสัญญาณวิทยุ การเรียกใช้มีอาร์กิวเมนต์ที่เกี่ยวข้องดังต่อไปนี้

module	มอดูลที่ต้องการใช้งาน
**destination*	พอยเตอร์หมายเลขโนดปลายทางที่ต้องการส่งข้อมูล
**msg*	พอยเตอร์ข้อมูล
msgLength	ขนาดความยาวข้อมูล (ไบต์)

int8_t readData(uint8_t module, uint8_t *msg, uint8_t *msgType, uint8_t msgLength);

ฟังก์ชันสำหรับอ่านข้อมูลที่รับมาจากเครื่องรับส่งสัญญาณวิทยุ การเรียกใช้มีอาร์กิวเมนต์ที่เกี่ยวข้องดังต่อไปนี้

module	มอดูลที่ต้องการใช้งาน
**msg*	พอยเตอร์ของข้อมูลที่รับ
**msgType*	พอยเตอร์ประเภทของข้อมูลที่รับ
msgLength	ขนาดความยาวข้อมูล (ไบต์)

4. คลาส PowerControlSystemC

ฟังก์ชันเริ่มการทำงาน

void begin();

ฟังก์ชันสำหรับสร้างการเชื่อมต่อไปยังวงจรตรวจวัดแบตเตอรี่และกำหนดค่าเริ่มต้นเปิดใช้งานวงจรพิเศษทุกวงจรในไมโครคอนโทรลเลอร์ ต้องเรียกใช้งานฟังก์ชันเป็นลำดับแรกเสมอก่อนการเรียกใช้ฟังก์ชันอื่น ๆ ภายในคลาสนี้

ฟังก์ชันจัดการพารามิเตอร์

int8_t writeParameter(uint8_t module, uint8_t paramName, uint8_t *value,
uint8_t valueLength);

ฟังก์ชันสำหรับกำหนดค่าพารามิเตอร์ของมอดูล สามารถเรียกใช้งานได้ตลอดทั้งโปรแกรมเมื่อต้องการเปลี่ยนแปลงค่าพารามิเตอร์ของมอดูลนั้น ฟังก์ชันนี้มีการคืนค่ากลับเป็นหมายเลขแทนข้อผิดพลาด การเรียกใช้มีอาร์กิวเมนต์ดังต่อไปนี้

module	มอดูลที่ต้องการใช้งาน
paramName	พารามิเตอร์ที่ต้องการเขียนค่า
**value*	พอยเตอร์ของข้อมูลที่ต้องการกำหนดค่าให้พารามิเตอร์
valueLength	ขนาดความยาวของข้อมูล (ไบต์)

int8_t readParameter(uint8_t module, uint8_t paramName, uint8_t *value,
uint8_t valueLength);

int8_t readParameter(uint8_t module, uint8_t paramName, float *value, uint8_t valueLength);

ฟังก์ชันสำหรับอ่านค่าพารามิเตอร์ของมอดูล สามารถเรียกใช้งานได้ตลอดทั้งโปรแกรมเมื่อต้องการอ่านค่าพารามิเตอร์ของมอดูลนั้น ฟังก์ชันนี้มีการคืนค่ากลับเป็นตัวเลขแทนผลของการทำงาน การเรียกใช้มีอาร์กิวเมนต์ที่เกี่ยวข้องดังต่อไปนี้

module	มอดูลที่ต้องการใช้งาน
paramName	พารามิเตอร์ที่ต้องการอ่านค่า
**value*	พอยเตอร์ของข้อมูลที่ต้องการเก็บค่าจากพารามิเตอร์
valueLength	ขนาดความยาวของข้อมูล (ไบต์)

ฟังก์ชันควบคุมการเปิดปิดมอดูล

int8_t enablePower(uint8_t module);

ฟังก์ชันสำหรับการเปิดใช้งานมอดูล สามารถเรียกใช้งานได้ตลอดทั้งโปรแกรม การทำงานของฟังก์ชันจะเป็นการสั่งเปิดการทำงานสำหรับมอดูลที่อ้างอิงกับไมโครคอนโทรลเลอร์และเปิดการจ่ายไฟฟ้าสำหรับมอดูลอื่น ๆ ฟังก์ชันนี้มีการคืนค่ากลับเมื่อทำงานเสร็จเป็นตัวเลขแสดงผลการทำงาน การเรียกใช้มีอากิวเมนต์ที่เกี่ยวข้องดังนี้

module	มอดูลที่ต้องการใช้งาน

int8_t disablePower(uint8_t module);

ฟังก์ชันสำหรับการปิดใช้งานมอดูล สามารถเรียกใช้งานได้ตลอดทั้งโปรแกรม การทำงานของฟังก์ชันจะเป็นการสั่งเปิดการทำงานสำหรับมอดูลที่อ้างอิงกับไมโครคอนโทรลเลอร์และปิดการจ่ายไฟฟ้าสำหรับมอดูลอื่น ๆ ฟังก์ชันนี้มีการคืนค่ากลับเมื่อทำงานเสร็จเป็นตัวเลขแสดงผลการทำงาน การเรียกใช้มีอากิวเมนต์ที่เกี่ยวข้องดังนี้

module	มอดูลที่ต้องการใช้งาน

ตารางรหัสตัวเลขแทนผลการทำงานของฟังก์ชัน

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

ประวัติผู้เขียนวิทยานิพนธ์

นายฐิติ สิทธิวางค์กูล สำเร็จการศึกษาวิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยธรรมศาสตร์ ในปีการศึกษา 2549 จากนั้นได้เข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตร์มหาบัณฑิต สาขาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2555

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University