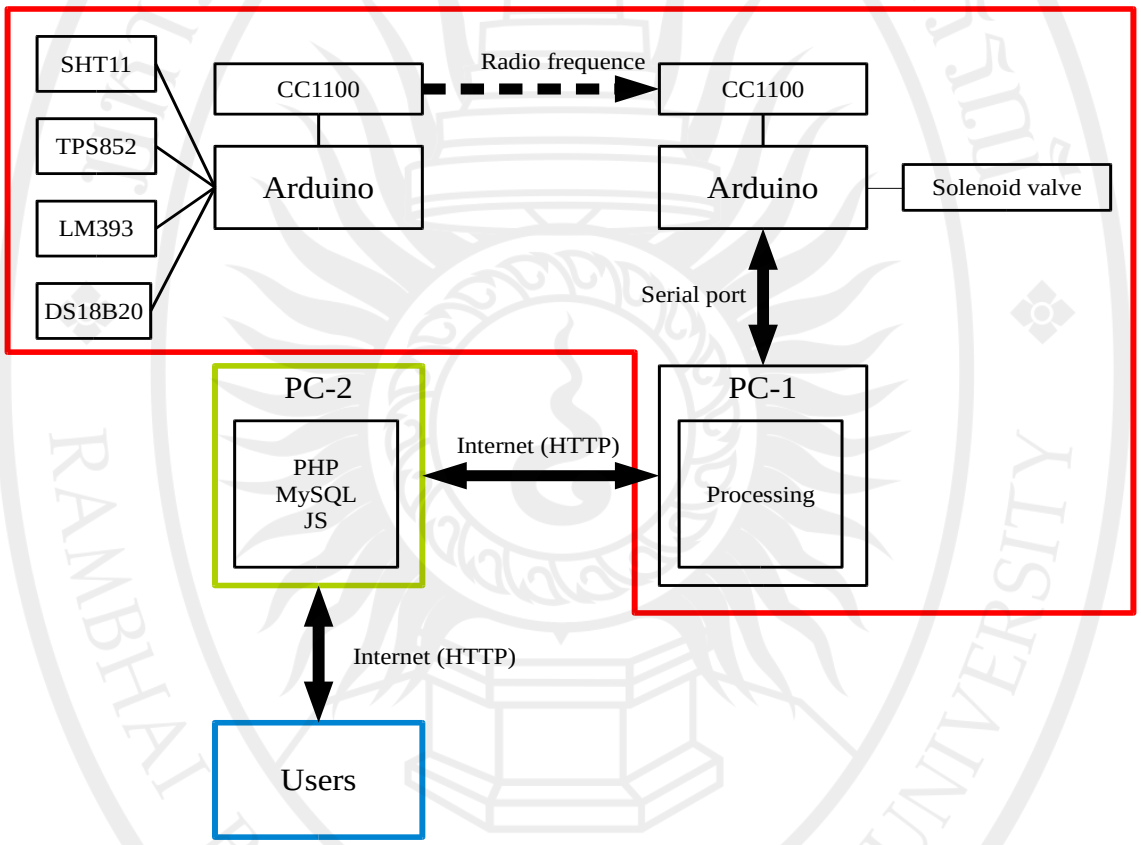
บทที่ 3
การวิเคราะห์และออกแบบระบบ

เนื้อหาในบทนี้จะเป็นการอธิบายถึงขั้นตอนการวิเคราะห์และออกแบบระบบ โดยระบบดังกล่าวนี้จะถูกแบ่งออกเป็นสองส่วนหลัก ๆ นั่นคือส่วนที่ทำหน้าที่ตรวจวัดข้อมูล ซึ่งจะทำหน้าที่เป็นส่วนลูกข่ายที่จะส่งข้อมูลไปยังอีกส่วนหนึ่งของระบบซึ่งทำหน้าที่แม่ข่ายภาพที่3.1 แสดงภาพรวมของระบบดังกล่าว



ภาพที่ 3.1 ภาพรวมของระบบที่พัฒนา (สีแดง) ระบบส่วนที่ติดตั้งที่สวน (สีเขียว) เครื่องแม่ข่ายกลาง และ (สีน้ำเงิน) ระบบฝั่งผู้ใช้

และเนื่องจากระบบงานนี้มีสองส่วนดังนั้นผู้วิจัยจึงจะขอแยกอธิบายรายละเอียดของระบบฝั่งลูกข่ายและฝั่งแม่ข่ายออกจากกัน โดยจะเริ่มจากระบบส่วนที่เป็นลูกข่าย (แสดงในกรอบสีแดง) ซึ่งเนื้อหาจะเกี่ยวกับกลุ่มของอุปกรณ์ที่จะทำหน้าที่ตรวจวัดและส่งค่าที่วัดได้ผ่านทางสัญญาณวิทยุไปให้เครื่องแม่ข่าย เนื้อหาในส่วนนี้จะรวมไปถึงการออกแบบโปรแกรมและเทคนิคที่ใช้ในการประสานงานระหว่างอุปกรณ์ต่าง ๆ ด้วย เนื้อหาในส่วนหลังของบทนี้จะเป็นการอธิบายเกี่ยวกับการออกแบบระบบที่ทำหน้าที่เป็นเครื่องแม่ข่าย (แสดงในกรอบสีเขียว) และส่วนของผู้ใช้งาน (แสดงในกรอบสีฟ้า) ซึ่งจะครอบคลุมถึงวิธีการส่งและ

รับข้อมูลผ่านสัญญาณวิทยุ,ระบบสำหรับบันทึกข้อมูลที่รับได้ และรวมถึงเทคนิคที่จะใช้ในการทำหน้าที่ควบคุมการเปิดปิดวาล์วน้ำ รายละเอียดเกี่ยวกับการออกแบบและพัฒนาระบบต่าง ๆ เหล่านี้มีดังต่อไปนี้

3.1 การออกแบบระบบฝั่งลูกข่าย

ดังที่ได้กล่าวมาแล้วว่าอุปกรณ์ต่าง ๆ ที่จัดเป็นกลุ่มอุปกรณ์ฝั่งลูกข่ายนั้นจะทำหน้าที่วัดค่าอุณหภูมิและความชื้นสัมพัทธ์ ทั้งของดินและอากาศ แล้วจึงส่งค่าที่วัดได้กลับไปยังเครื่องแม่ข่ายในรูปของสัญญาณวิทยุจากแนวคิดดังกล่าวนี้เราจะพบว่านอกจากเซนเซอร์ที่จะต้องใช้เพื่อวัดค่าเหล่านี้และตัวรับส่งวิทยุแล้ว เราจะต้องมีอุปกรณ์อีกหนึ่งชิ้นที่มาจะทำหน้าที่ประสานงานและควบคุมการทำงานของเซนเซอร์และตัวรับส่งวิทยุเหล่านั้น ซึ่งโดยปกติแล้วเรานิยมใช้ไมโครคอนโทรเลอร์มาทำหน้าที่ประสานงานดังกล่าว ดังนั้นงานวิจัยนี้จึงได้กำหนดใช้บอร์ด Arduino เป็นอุปกรณ์หลักในการทำหน้าที่ควบคุมการทำงานระหว่างเซนเซอร์และตัวรับส่งวิทยุ ซึ่งจะรวมถึงการแปลงข้อมูลจากเซนเซอร์ให้อยู่ในรูปแบบที่เหมาะสมกับการส่งสัญญาณวิทยุอีกด้วยดังนั้นเนื้อหาในส่วนนี้จะเป็นการอธิบายรายละเอียดเกี่ยวกับวิธีการทำงานกับบอร์ด Arduino, เซนเซอร์ และตัวรับส่งวิทยุ รวมถึงแนวคิดในการการออกแบบโปรแกรมที่จะใช้ควบคุมการทำงานต่าง ๆ ดังรายละเอียดต่อไปนี้

3.1.1 บอร์ด Arduino

Arduino[1]เป็นชื่อของบอร์ดไมโครคอนโทรลเลอร์ที่มีขนาดเล็กและมีราคาถูก ที่มีการพัฒนาขึ้นมาบนแนวคิดแบบโอเพนซอร์ซ (open source) นั่นคือผู้พัฒนาจะเปิดเผยข้อมูลต้นฉบับทั้งส่วนของฮาร์แวร์และซอฟต์แวร์ที่ใช้ในการทำงาน ภาพที่ 3.2 แสดงตัวอย่างของบอร์ด Arduino รุ่น Uno

Arduino Uno Arduino Uno R2 Arduino Uno R3

ภาพที่ 3.2 ตัวอย่างบอร์ด Arduinoรุ่น UNO ทั้งสามแบบ

ที่มา: Arduino., 2014

1 http://www.arduino.cc/

บอร์ด Arduino นี้ถูกพัฒนาขึ้นมาในช่วงปี ค.ศ. 2005และปัจจุบันอยู่ภายใต้การจัดการของบริษัท Arduino และเนื่องจาก Arduino เป็นบอร์ดไมโครคอนโทรเลอร์แบบโอเพนซอร์ซดังนั้นจึงมีผู้นำบอร์ด Arduino นี้ไปต่อยอดดัดแปลงความสามารถเพิ่มเติมได้เป็นอีกหลายกลุ่มย่อย แต่โดยรวมนั้นบอร์ด Arduino ส่วนใหญ่จะใช้หน่วยประมวลผลแบบ 8 บิตของ Atmel AVR หรืออาจจะมีบางรุ่นที่ใช้หน่วยประมูวลผูลแบบ 32 บิตของ Atmel ARM และโดยทั่วไปแล้วบอร์ด Arduino จะมีขา(pin) สัญญาณแบบจีพีไอโอ (general-purpose input/output : GPIO) ไว้ให้ผู้ใช้เพื่อใช้ในการเชื่อมต่อกับอุปกรณ์ภายนอก โดยขาจีพีไอโอนี้จะรองรับการสื่อสารทั้งแบบแอนาล็อกและดิจิทัล โดยผู้ใช้สามารถควบคุมการทำางานของบอร์ดผ่านการเขียนโปรแกรมด้วยภาษาที่มีไวยกรณ์คล้ายกับภาษากลุ่ม C/C++ และภาษาจาวาโดยโปรแกรมเมอร์จะพัฒนาโดยใช้ไอดีอี (integrated development editor : IDE) ที่ชื่อว่า Arduino IDE ซึ่งเป็นโปรแกรมแบบโอเพนซอร์ซเช่นเดียวกับตัวบอร์ดตารางที่3.1 เปรียบเทียบคุณสมบัติของบอร์ด Arduino รุ่นต่างๆ ตามข้อมูลที่ปรากฏในปีพ.ศ. 2558

ตารางที่ 3.1 คุณสมบัติของบอร์ด Arduino รุ่นต่าง ๆ (ข้อมูลปีพ.ศ.2558)
ที่มา: arduino,2014

รุ่น	Processor	Operating voltage (V)	Input voltage (V)	CPU speed (MHz)	Analog In/Out	Digital In/Out	EEPROM (KB)	SRAM (KB)	Flash (KB)	UART
Uno	ATmega328	5	7-12	16	6/0	14/6	1	2	32	1
Due	AT91SAM3X8E	3.3	7-12	84	12/2	54/12	-	96	512	4
Lenonardo	ATmaga32u4	5	7-12	16	12/0	20/7	1	2.5	32	1
Mega 2560	ATmega2560	5	7-12	16	16/0	54/15	4	8	256	4
Mega ADK	ATmega2560	5	7-12	16	16/0	54/15	4	8	256	4
Micro	ATmega32u4	5	7-12	16	12/0	20/7	1	2.5	32	1
Mini	ATmega328	5	7-9	16	8/0	14/6	1	2	32	-
Nano	ATmega168 ATmega328	5	7-9	16	8/0	14/6	0.512,1	1,2	16,32	1
Ethernet	ATmega328	5	7-12	16	6/0	14/4	1	2	32	-
Esplora	ATmega32u4	5	7-12	16	-	-	1	2	32	-
ArduinoBT	ATmega328	5	2.5-12	16	6/0	14/6	1	2	32	1
Fio	ATmega328	3.3	3.7-7	16	6/0	14/6	1	2	32	1

ตารางที่ 3.1 คุณสมบัติของบอร์ด Arduino รุ่นต่าง ๆ (ต่อ)

รุ่น	Processor	Operating voltage (V)	Input voltage (V)	CPU speed (MHz)	Analog In/Out	Digital In/Out	EEPROM (KB)	SRAM (KB)	Flash (KB)	UART
Pro (168)	ATmega168	3.3	3.35-12	8	6/0	14/6	0.512	1	16	1
Pro (328)	ATmega328	5,3.3	5-12	16	6/0	14/6	1	2	32	1
Pro Mini	ATmega168	3.3,5	3.35-12,5-12	8,16	6/0	14/6	0.512	1	16	1
LilyPad	ATmega168V ATmaga328V	2.7-5.5	2.7-5.5	8	6/0	14/6	0.512	1	16	-
LilyPad USB	ATmega32u4	3.3	3.8-5	8	4/0	9/4	1	2.5	32	-
LilyPad Simple	ATmega328	2.7-5.5	2.7-5.5	8	4/0	9/4	1	2	32	-
LillyPad SimpleSnap	ATmega328	2.7-5.5	2.7-5.5	8	4/0	9/4	1	2	32	-
Yun	ATmega32u4	5		16	12/0	20/7	1	2.5	32	1

สำหรับงานวิจัยนี้ ผู้วิจัยนี้ได้เลือกใช้บอร์ดArduinoรุ่น UNO เนื่องจากเป็นบอร์ดที่มีราคา, จำนวนส่วนเชื่อมต่ออุปกรณ์ภายนอก และประสิทธิภาพเหมาะสมกับการวิจัยนี้จากตารางที่ 3.1 จะเห็นว่าบอร์ด Arduino UNO เป็นบอร์ดไมโครคอนโทรเลอร์ที่ใช้ชิปรุ่น ATmega328 ที่มีความเร็ว 16 เมกะเฮิรตซ์เป็นหน่วยประมวลผลหลัก บอร์ดรุ่นนี้จะมีขาแอนะล๊อก (analog pin) สำหรับใช้ในการอ่านข้อมูลจำนวนหกขา และมีขาดิจิทัล (digital pin) ที่สามารใช้ได้ทั้งการรับและส่งข้อมูลอีกจำนวน 14 ขา โดยในจำนวนขาดิจิทัลเหล่านี้จะมีขาที่สามารถใช้เพื่อการส่งสัญญาณแบบมอดูเลตเชิงความว้างพัลส์ (pulse width modulation : PWM) ได้ทั้งหมดหกขา บอร์ดนี้มีหน่วยความจำแบบไม่ลบเลือน (non-volatile memory) จำนวนสองชุด คือ อีอีพร็อม (electrically erasable PROM : EEPROM) และแฟลช (flash) ที่สามารถใช้บันทึกข้อมูลจำนวน 1 และ 32 กิโลไบต์ตามลำดับ โดยเราจะใช้หน่วยความจำแฟลชเป็นพื้นที่หลักสำหรับจัดเก็บโปรแกรมที่พัฒนาขึ้น นอกจากนั้นแล้วบอร์ด Arduino UNO ยังมีหน่วยความจำแบบลบเลือนได้ (volatile memory) คือเอสแรม (static RAM : SRAM) อีกจำนวน 2 กิโลไบต์ จุดแตกต่างประการหนึ่งระหว่างบอร์ด Arduino UNO กับบอร์ด Ardunio รุ่นอื่น ๆ ก็คือบอร์ด Arduino UNO ไม่ได้ใช้ไดรฟเวอร์FTDI USB-to-serial ในการสื่อสาร แต่จะใช้ Atmega16U2 ในการทำงานดังกล่าวแทน โดยปัจจุบันบอร์ด Arduino UNO ได้พัฒนามาถึงรุ่นปัจจุบันคือ Arduino UNO Revision 3 และตารางที่ 3.2 แสดงข้อมูลเพิ่มเติมเกี่ยวกับบอร์ด Arduino UNO ที่จะใช้ในการวิจัยครั้งนี้

ตารางที่ 3.2 คุณสมบัติของบอร์ด Arduino UNO

ที่มา: arduino,2014

Microcontroller	ATmega328
Operating voltage	5V
Input voltage (recommended)	7-12V
Input voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O pin	40mA
DC Current for 3.3V pin	50mA
Flash Memory	32KB (ATmega328) of which 0.5KB used by bootloader
SRAM	2KB (ATmega328)
EEPROM	1KB (Atmega328)
Clock Speed	16MHz
Length	68.6mm
Width	53.4mm
Weight	25g

3.1.2 เซนเซอร์วัดอุณหภูมิและความชื้นสัมพันธ์ของอากาศ

สำหรับอุปกรณ์ที่จะมาทำหน้าที่วัดอุณหภูมิและความชื้นสัมพัทธ์ของอากาศนั้น งานวิจัยนี้เลือกใช้ SHT11 ของบริษัท Sensirion ซึ่งเป็นชิปแบบเอสเอ็มดี(surface mounted device :SMD) โดยที่ชิปนี้ได้รวบเซนเซอร์สำหรับการวัดค่าอุณหภูมิและความชื้นสัมพัทธ์เข้าไว้ในตัวเดียวกันชิป SHT11 นี้จะใช้เซนเซอร์ชนิดเก็บประจุ(capacitive sensor)ในการวัดความชื้นสัมพัทธ์และใช้เซนเซอร์แบบช่องว่างแถบพลังงาน(band-gap sensor) ในการวัดอุณหภูมิ และอุปกรณ์นี้ใช้เทคโนโลยี CMOSens เพื่อเพิ่มความเสถียรของการทำงานในระยะยาว นอกจากนั้นแล้วชิปนี้ยังมีหน่วยประมวลผลสัญญาณดิจิทัลด้วย จึงทำให้อุปกรณ์นี้สื่อสารกับไมโครคอนโทรเลอร์ได้แบบดิจิทัลซึ่งจะช่วยเพิ่มความแม่นยำของค่าที่อ่านได้ โดยผู้ใช้สามารถกำหนดความละเอียดของค่าที่วัดได้ถึง 14 บิต ข้อมูลเกี่ยวกับชิปSHT11ในการวัดความชื้นสัมพัทธ์และอุณหภูมิของอากาศแสดงอยู่ในตารางที่ 3.3 และ 3.4 ตามลำดับ

ตารางที่ 3.3 คุณสมบัติของ SHT11 ในการวัดความชื้นสัมพัทธ์ของอากาศ

ที่มา: arduino,2014

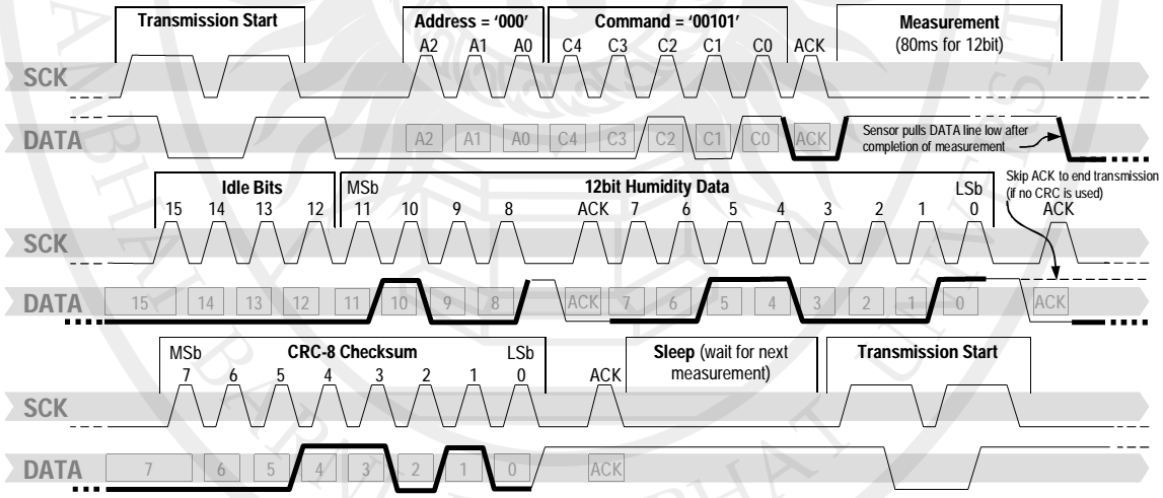
Paramter	Condition	min	typ	max	Units
Resolution		0.4	0.05	0.05	%RH
		8	12	12	bit
Accuracy SHT11	typical		±3.0		%RH
	maximal	see Figure 2			
Repeatability			±0.1		%RH
Hysteresis			±1		%RH
Non-linearity	linearized		<<1		%RH
Response time	τ(63%)		8		s
Operating Range		0		100	%RH
Long term drift	normal		<0.5		%RH/yr

ตารางที่ 3.4 คุณสมบัติของ SHT11 ในการวัดอุณหภูมิของอากาศ

ที่มา: arduino,2014

Paramter	Condition	min	typ	max	Units
Resolution		0.04	0.01	0.01	ºC
		12	14	14	bit
Accuracy SHT11	typical		±0.4		ºC
	maximal	see Figure 3			
Repeatability			±0.1		ºC
Operating Range		-40		123.8	ºC
		-40		254.9	ºF
Response Time	τ(63%)	5		30	s
Longterm drift			<0.04		ºC/yr

ชิป SHT11 จะสื่อสารกับไมโครคอนโทรเลอร์ด้วยโพรโทคอล I2C (inter-integrated circuit) ซึ่งเป็นโพรโทคอลที่พัฒนาโดยบริษัท Philips Semicondutor โดยได้รับการออกแบบมาให้ใช้ขาเพื่อส่งสัญญาณจำนวนสองขา (DATA และ SCK) และส่งกระแสไฟอีกสองขา (VDD และ GND) รวมกันทั้งหมดสี่ขาและมีขั้นตอนหลัก ๆ สำหรับใช้ในการติดต่อสื่อสารระหว่างอุปกรณ์กับไมโครคอนโทรเลอร์ดังต่อไปนี้ ขั้นแรกคือการสั่งให้เซนเซอร์เริ่มทำงานซึ่งทำได้โดยการที่ไมโครคอนโทรเลอร์ส่งกระแสไฟเข้าไปยังเซนเซอร์ทางขา VDD แล้วรอสักครู่ (เช่นประมาณ11มิลลิวินาที) เพื่อให้อุปกรณ์ได้เตรียมความพร้อมในการทำงานให้เรียบร้อยก่อนที่จะเข้าสู่โหมดหลับ (sleep mode) เมื่อเซนเซอร์พร้อมทำงานแล้ว (อยู่ในโหมดหลับ) ไมโครคอนโทรเลอร์จะสามารถสั่งให้เซนเซอร์วัดค่าที่ต้องการได้โดยการส่งคำสั่งงาน (command) ไปยังอุปกรณ์ โดยที่คำสั่งเหล่านี้จะอยู่ในรูปของสัญญาณที่เข้าไปยังอุปกรณ์ทางขา DATA และควบคุมการประสานเวลา (synchronize) การรับส่งบิตด้วยสัญญาณที่ขา SCK และเมื่ออุปกรณ์รับคำสั่งงานเรียบร้อยแล้วก็จะทำงานตามคำสั่งนั้น ในกรณีที่เป็นการสั่งวัดค่า อุปกรณ์ก็จะทำการวัดค่าแล้วบันทึกค่าที่วัดได้ลงในเรจิสเตอร์ของอุปกรณ์ดังนั้นไมโครคอนโทรเลอร์จะต้องรอสักครู่เพื่อให้อุปกรณ์ทำงานตามที่กำหนดไว้ให้เสร็จเรียบร้อยก่อนจึงจะสามารถอ่านค่าที่ออกมาจากเรจิสเตอร์ได้ ภาพที่ 3. 3 แสดงตัวอย่างการสื่อสารระหว่างSHT11กับไมโครคอนโทรเลอร์ โดยภาพนี้แสดงขั้นตอนการส่งคำสั่งงานเริ่มต้นส่งสัญญาณ(transmission start command), การวัดความชื้นสัมพัทธ์,การอ่านค่าและจบการทำงานตามลำดับ



ภาพที่ 3.3 ตัวอย่างการสื่อสารระหว่าง SH11 กับไมโครคอนโทรเลอร์

ที่มา: Sensirion, 2010

3.1.3 เซนเซอร์วัดอุณหภูมิของดิน

สำหรับเซนเซอร์ที่จะใช้เพื่อวัดอุณหภูมิของดินนั้นผู้วิจัยเลือกใช้ชิปDS18B20 ซึ่งเป็นเทอร์โมมิเตอร์แบบดิจิทัลที่สามารถให้ความละเอียดของค่าที่วัดได้ระหว่าง 9 - 12 บิต สามารถทำงานได้ในช่วงอุณหภูมิ -55 ถึง125 องศาเซลเซียส และมีความแม่นยำของค่าที่วัดได้อยู่ที่ประมาณ +-0.5 องศา

เซลเซียส ในช่วง -10 ถึง 85 องศาเซลเซียส

อุปกรณ์นี้จะสื่อสารกับไมโครคอนโทรเลอร์โดยใช้โพรโทคอล 1-Wire โดยที่ 1-Wire นี้เป็นโพรโตคอลที่ออกแบบโดยบริษัท Dallas Semiconductor Corp เพื่อใช้สำหรับการส่งข้อมูลความเร็วต่ำและสามารถส่งสัญญาณรวมทั้งกระแสไฟโดยโพรโทคอลนี้มีจุดเด่นอยู่ที่การใช้สายเพียงเส้นเดียว โพรโทคอลนี้มีหลักการทำงานคล้ายคลึงกับโพรโทคอล I2C ซึ่งในกรณีของโพรโทคอล1-Wire นั้นจะทำงานด้วยอัตราการส่ง/รับข้อมูล (data rate) ที่ต่ำกว่าโพรโทคอล I2C แต่โพรโทคอล1-Wire นั่นจะสามารถส่งข้อมูลได้เป็นระยะทางที่ไกลกว่าโพรโทคอล I2C นอกจากนั้นแล้วชิป DS18B20 แต่ละตัวจะมีรหัสประจำตัวที่ไม่ซ้ำกันเลย ดังนั้นข้อดีประการหนึ่งของการสื่อสารด้วยโพรโทคอล 1-Wire ก็คือผู้ใช้สามารถใช้ไมโครคอนโทรเลอร์สื่อสารกับเซนเซอร์นี้ได้หลาย ๆ ตัวโดยใช้สายสัญญาณเพียงเส้นเดียว

3.1.4 เซนเซอร์วัดความชื้นสัมพัทธ์ของดิน

งานวิจัยนี้จะวัดความชื้นสัมพัทธ์ของดินด้วยเทคนิคการวัดความจุไฟฟ้าของดินโดยใช้อุปกรณ์ซึ่งสามารถวัดปริมาณน้ำในดินจากการวัดการนำไฟฟ้า ดังนั้นงานวิจัยนี้จึงเลือกใช้ LM393 เป็นอุปกรณ์หลักเนื่องจากเป็นอุปกรณ์ที่มีทำงานได้ตามความต้องการและมีราคาถูก ทั้งนี้ชิป LM393 จะสื่อสารกับไมโครคอนโทรเลอร์แบบแอนะล็อกดังนั้นไมโครคอนโทรเลอร์จะสามารถอ่านค่าความชื้นสัมพัทธ์ของดินที่วัดได้โดยการอ่านข้อมูลจากขาแอนะล็อกได้โดยตรง (ในกรณีของบอร์ด Ardino UNO นั้นจะมีขาแอนะล็อกจำนวนหกขา จึงสามารถต่อพ่วงอุปกรณ์นี้ได้สูงสุดถึงหกตัว) โดยบอร์ด Arduino UNO มีความละเอียดของค่าที่อ่านจากขาแอนะล็อกอยู่ที่ 10 บิต จึงทำให้ข้อมูลดิบ (raw data) ที่เราจะอ่านได้จากขาแอนะล็อกนี้เป็นเลขจำนวนเต็มที่มีค่าระหว่าง 0 ถึง 1023 ทั้งนี้บอร์ด ArduinoUNO จะใช้เวลาในการอ่านข้อมูลจากขาแอนะล็อกประมาณ 100 ไมโครวินาที (ปัญหาจากการอ่านข้อมูลจากขาแอนะล็อกและวิธีการจัดการปัญหานี้จะกล่าวถึงในบทต่อไป)ตัวอย่างของเซนเซอร์นี้แสดงในภาพที่ 3.4 (ซ้าย)

LM393 TPS852

ภาพที่ 3.4 เซนเซอร์วัดข้อมูลอากาศ (ซ้าย) LM393 สำหรับวัดความชื้นสัมพัทธ์ และ (ขวา) TPS85 เซนเซอร์วัดอุณหภูมิอากาศ

3.1.5 เซนเซอร์วัดความเข้มแสงแวดล้อม

ผู้วิจัยเลือกใช้ TPS852 เป็นเซนเซอร์วัดความเข้มแสงแวดล้อม (ambient light intensity) โดยที่ TPS852 เป็นเซนเซอร์แบบเอสเอ็มดีที่ผลิตโดยบริษัท TOSHIBA สำหรับใช้วัดความเข้มแสงแวดล้อมอุปกรณ์นี้ได้รวมโฟโตไดโอด (photodiode) และวงจรกระแสเข้าไว้ในชิปตัวเดียวข้อดีประการหนึ่งของเซนเซอร์นี้คือการเป็นอุปกรณ์แบบเอสเอ็มดีที่มีขนาดเล็กมาก (อุปกรณ์นี้มีขนาดเพียง 1.6x1.6x0.55 มิลลิเมตรเท่านั้น) นอกจากนั้นแล้วเซนเซอร์นี้ยังใช้กระแสต่ำมากอีกด้วย ดังนั้นอุปกรณ์นี้จึงเหมาะกับการใช้ในงานวิจัยนี้เป็นอย่างมากด้วยเช่นกัน ทั้งนี้อุปกรณ์นี้จะสื่อสารกับไมโครคอนโทรเลอร์ด้วยสัญญาณแบบแอนะล็อกเช่นเดียวกับ LM393 ตัวอย่างของเซนเซอร์นี้แสดงในภาพที่ 3.4

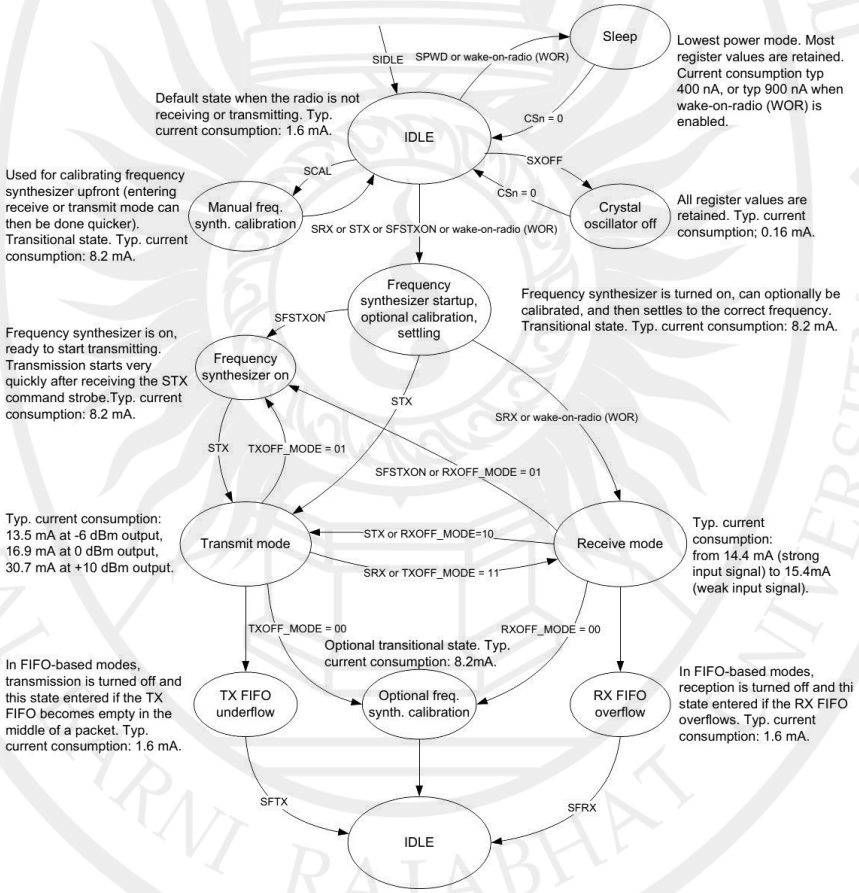
3.1.6 อุปกรณ์รับส่งสัญญาณวิทยุ

ข้อมูลที่วัดได้จากเซนเซอร์ดังกล่าวทั้งหมดจะส่งกลับไปยังเครื่องแม่ข่ายโดยใช้สัญญาณวิทยุเป็นสื่อ ดังนั้นงานวิจัยนี้จึงเลือกใช้ชิป CC1100 เป็นอุปกรณ์หลักในการสื่อสารระหว่างเครื่องแม่ข่ายกับไมโครคอนโทรเลอร์ โดยที่ CC1100 นี้เป็นชิปที่ทำหน้าที่เป็นเครื่องรับส่งวิทยุ (transceiver) ของบริษัท Teaxs Instrument โดยมีความสามารถในการสื่อสารด้วยความถี่ในช่วง 300-348,400-464 และ 800-928 เมกะเฮิรตซ์และมีอัตราบอด (baud rate) ได้สูงสุดประมาณ 500 กิโลบอด (kbaud)

ภาพที่ 3.5 ตัวอย่าง CC1100

อุปกรณ์นี้จะสื่อสารข้อมูลกับไมโครคอนโทรเลอร์โดยใช้โพรโทคอล SPI (serial peripheral interface) ซึ่งเป็นโพรโทคอลที่ออกแบบมาสำหรับการสื่อสารแบบอนุกรมแบบประสานเวลา (synchronous)โดยหลักการแล้วโพรโทตอลเอสพีไอจะสื่อสารกับไมโครคอนโทรลเลอร์โดยการใช้ขาสัญญาณจำนวนสี่ขา (ไม่นับรวมขา VDD และ GND) ประกอบด้วย MOSI, MISO, SCLK และ SS เพื่อทำหน้าที่ต่างๆ ดังต่อไปนี้

- **MOSI**หรือ master output/slave input ขาสัญญาณนี้จะทำหน้าที่รับสัญญาณที่ส่งมาจากไมโครคอนโทรเลอร์ (เรียกว่าmaster) มาเข้าสู่อุปกรณ์ (เรียกว่า slave)
- **MISO**หรือ master input/slave output จะทำหน้าที่รับสัญญาณจาก slave ส่งไปยัง master
- **SCLK**หรือ serial clock เป็นขาสัญญาณนาฬิกาเพื่อการประสานจังหวะการทำงานที่ส่งมาจากไมโครคอนโทรเลอร์
- **SS**หรือ chip select เป็นขาสัญญาณสำหรับการเลือก slave (ในกรณีที่มีการเชื่อมต่ออุปกรณ์หลายตัวเข้ากับไมโครคอนโทรเลอร์ตัวเดียว)



ภาพที่ 3.6 สเตทไดอะแกรมของ CC1100

ที่มา: Texas Instrument, 2014

โดยกรณีทั่วไปนั้น เราจะกำหนดให้CC1000 ทำหน้าที่เป็น slave และให้ไมโครคอนโทรเลอร์ทำหน้าที่เป็น master โดยเราจะต้องใช้โพรโทคอล SPI นี้เพื่อการเขียนและอ่านข้อมูลจากเรจิสเตอร์ของอุปกรณ์อีกด้วยเมื่อไมโครคอนโทรเลอร์ต้องการจะสื่อสารกับ CC1100 นั้นไมโครคอนโทรเลอร์จะต้องส่ง

คำสั่งงานซึ่งมีข้อมูลส่วนหัว (header) ไปให้อุปกรณ์ ข้อมูลส่วนหัวนี้จะประกอบด้วยบิต R/W และบิตที่อยู่ (address bit)ของอุปกรณ์ และไมโครคอนโทรเลอร์จะต้องทำให้ระดับสัญญาณที่ขา SS มีค่าเป็น LOW ตลอดเวลาที่มีการรับ/ส่งข้อมูลด้วยไพรโทคอล SPI อีกด้วย ภาพที่ 3.6 แสดงสเตทไดอะแกรมของโหมดการทำงานของ CC1100 และตารางที่ 3.5แสดงเวลาที่ต้องใช้ในการดำเนินการแต่ละขั้นตอน

ตารางที่ 3.5 เวลาที่จำเป็นต้องใช้ในการสื่อสารด้วยโพรโทคอล SPI ของ CC1100
ที่มา: Texas Instruments, 2014

Parameter	Description	Min	Max	Units
f_{SCLK}	SCLK frequency 100 ns delay inserted between address byte and data byte (single access), or between address and data,-and between each data byte (burst access). SCLK frequency, single access No delay between address and data byte SCLK frequency, burst access No delay between address and data byte, or between data bytes	- - - 	10 9 6.5 	MHz
$t_{sp,pd}$	CSn low to positive edge on SCLK, in power-down mode	150	-	µs
t_{sp}	CSn low to positive edge on SCLK, in active mode	20	-	ns
t_{ch}	Clock high	50	-	ns
t_{cl}	Clock low	50	-	ns
t_{rise}	Clock rise time	-	5	ns
t_{fall}	Clock fall time	-	5	ns
t_{sd}	Setup data (negative SCLK edge) to positive edge on SCLK (t_{sd}applies between address and data bytes, and between data bytes)	55 76	- -	ns
t_{hd}	Hold data after positive edge on SCLK	20	-	ns
t_{ns}	Negative edge on SCLK to CSn high.	20	-	ns

ในระหว่างการทำงานอุปกรณ์จะบอกให้ไมโครคอนโทรเลอร์รับรู้สถานะการทำงานของตัวเอง ด้วยการบันทึกสถานะปัจจุบันของอุปกรณ์ไว้ในเรจิสเตอร์ ซึ่งไมโครคอนโทรเลอร์สามารถใช้ข้อมูลนี้เพื่อติดตามผลการปฏิบัติงานตามคำสั่งงานต่างๆ ได้ข้อมูลนี้จะมีขนาดหนึ่งไบต์และ เราจะเรียกไบต์นี้ว่าไบต์สถานะชิป (chip status byte) ซึ่งเป็นไบต์ส่วนหัวอีกชนิดหนึ่งโดยข้อมูลในไบต์นี้จะแบ่งออกเป็นสามกลุ่ม ดังนี้

- บิตที่หนึ่งถึงสี่ ข้อมูลในสี่บิตนี้จะบอกสถานะของบัฟเฟอร์ในการส่งและรับข้อมูล
- บิตที่ห้าถึงเจ็ด ข้อมูลกลุ่มนี้คือข้อมูลสถานะของอุปกรณ์
- บิตที่แปด เป็นบิตแสดงสถานะพร้อมทำงาน (chip ready) ซึ่งบิตนี้จะบอกให้ไมโครรู้ว่าอุปกรณ์กำลังทำงานอยู่

รายละเอียดเกี่ยวกับไบต์สถานะนี้แสดงอยู่ในตารางที่ 3.6

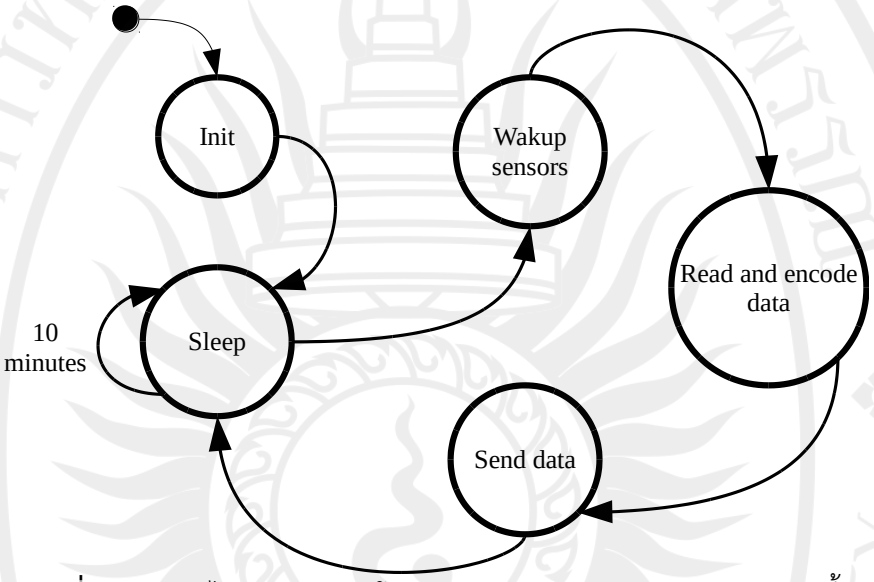
ตารางที่ 3.6 ไบต์สถานะของ CC1100

ที่มา: Texas Instruments, 2014

Bits	Name	Description		
7	CHIP_RDYn	Stays high until power and crystal have stabilized. Should always be low when using the SPI interface.		
6:4	STATE[2:0]	Indicates the current main state machine mode		
		Value	State	Description
		000	IDLE	IDLE state (Also reported for some transitional states instead of SETTLING or CALIBRATE)
		001	RX	Recevie mode
		010	TX	Transmit mode
		011	FSTXON	Fast TX ready
		100	CALIBRATE	Frequency synthesizer calibration is running
		101	SETTLING	PLL is settling
		110	RXFIFO_OVERFLOW	RX FIFO has overflowed. Red out any usefuldata,then flush the FIFO with SFRX
		111	TXFIFO_UNDERFLOW	TX FIFO has underflowed. Acknowledge with SFTX
3:0	FIFO_BYTES_AVAILABLE[3:0]	The number of bytes available in the RX FIFO or free bytes in the TX FIFO		

3.1.7 การควบคุมการทำงานด้วยไมโครคอนโทรเลอร์

ดังที่ได้กล่าวมาแล้วว่างานวิจัยนี้จะใช้บอร์ด Arduino รุ่น UNO ในการควบคุมการทำงานของเซนเซอร์ต่างๆ บอร์ด Arduino UNO นี้จะทำหน้าที่ทั้งการเตรียมความพร้อมการใช้งาน, การสั่งวัดค่า, การอ่านค่าที่วัดได้ รวมไปถึงการควบคุมการส่งสัญญาณวิทยุไปยังเครื่องแม่ข่าย ดังนั้นเนื้อหาในหัวข้อส่วนนี้จะแสดงรายละเอียดเกี่ยวกับการออกแบบโปรแกรมบนบอร์ด Arduino เพื่อทำหน้าที่ต่างๆ ดังกล่าว



ภาพที่ 3.7 สแตทไดอะแกรมของโปรแกรมควบคุมการทำงานของงานวิจัยนี้

ขั้นตอนแรกของการทำงานของบอร์ด Arduino คือ การเตรียมความพร้อมของอุปกรณ์ที่เชื่อมต่ออยู่ โดยหลังจากที่บอร์ด Arduino พร้อมทำงานแล้ว โปรแกรมจะไปกำหนดให้ไมโครคอนโทรเลอร์ส่งสัญญาณไปยังอุปกรณ์ต่าง ๆ ที่เชื่อมต่ออยู่เพื่อให้อุปกรณ์เหล่านั้นเริ่มต้นทำงาน ซึ่งขั้นตอนการเตรียมความพร้อมดังกล่าวนี้ ส่วนใหญ่จะเป็นการกำหนดค่าการทำงานเบื้องต้น ในกรณีของเซนเซอร์นั้นอาจจะเป็นการกำหนดค่าพารามิเตอร์ต่าง ๆ ของการวัด เช่น การกำหนดความละเอียดของค่าที่ต้องการวัดและสำหรับกรณีของเครื่องรับส่งวิทยุนั้นจะเป็นการกำหนดค่าถี่ของสัญญาณวิทยุที่จะใช้ในการสื่อสาร, รูปแบบการใช้พลังงานรวมถึงการกำหนดรูปแบบข้อมูลที่จะใช้ตรวจสอบข้อมูลที่จะส่ง เป็นต้น หลังจากกำหนดค่าต่าง ๆ เรียบร้อยแล้วบอร์ด Arduino ก็เข้าเข้าสู่ลูปหลัก โดยงานวิจัยนี้กำหนดให้บอร์ด Arduino มีงานที่จะต้องทำในลูปหลักสามอย่าง คือ การวัดค่า, การส่งสัญญาณวิทยุ และการเข้าสู่โหมดประหยัดพลังงาน

โดยงานหลักส่วนใหญ่ของไมโครคอนโทรเลอร์จะเป็นการสั่งวัดค่าและอ่านค่าจากเซนเซอร์ต่าง ๆ ซึ่งผู้วิจัยได้แยกอ่านข้อมูลแยกเซนเซอร์ที่สื่อสารแบบดิจิทัลก่อน หลังจากนั้นจึงอ่านข้อมูลจากเซนเซอร์ที่สื่อสารแบบแอนะล็อก ทั้งนี้เพราะการอ่านข้อมูลจากขาแอนะล็อกนั้นจะต้องมีการหน่วงเวลาเล็กน้อยเพื่อให้เอดีซี (analog-to-digital converter) ปรับระดับแรงดันไฟฟ้าที่ขาแอนะล็อกให้อยู่ในระดับที่

เหมาะสมก่อนจึงจะอ่านค่าได้อย่างไรก็ตามข้อมูลที่อ่านได้จากเซนเซอร์ต่าง ๆ เหล่านี้มีหลายรูปแบบ โดยข้อมูลที่อ่านได้จากขาแอนะล็อกทั้งหมดจะเป็นค่าเลขจำนวนเต็มแบบ 10 บิต (มีค่าระหว่าง 0 ถึง 1023) ในขณะที่ค่าที่อ่านได้จากเซนเซอร์ที่สื่อสารแบบดิจิทัลนั้นจะให้ผลลัพธ์เป็นเลขจำนวนจริงที่ค่อนข้างพร้อมใช้งานได้ทันที ดังนั้นหลังจากที่อ่านข้อมูลจากเซนเซอร์ต่างๆ แล้ว ไมโครคอนโทรเลอร์จึงจะต้องนำข้อมูลที่อ่านได้เหล่านี้มาเข้ารหัสข้อมูลเพื่อให้เป็นข้อมูลรูปแบบเดียวกันและเพื่อให้เหมาะสมกับรูปแบบการส่งข้อมูลด้วยสัญญาณวิทยุ โดยในกรณีของงานวิจัยนี้จะกำหนดให้แปลงข้อมูลทั้งหมดให้อยู่ในรูปแบบของแถวลำดับของไบต์ (byte array) เพื่อความสะดวกในการส่งข้อมูลด้วย CC1100 โดยได้กำหนดให้ข้อมูลดังกล่าวเป็นแถวลำดับที่มีสมาชิกจำนวน 13 ไบต์(รายละเอียดจะได้กล่าวถึงในบทต่อไป)

ขั้นตอนต่อมาหลังจากแปลงข้อมูลให้อยู่ในรูปแบบของแถวลำดับของไบต์เรียบร้อยแล้ว ไมโครคอนโทรเลอร์จะปลุก CC1100 ให้ตื่นจากโหมดหลับมาอยู่ในโหมดเดินเครื่องเปล่า (idle) และพร้อมทำงาน หลังจากนั้นแล้วจึงนำแถวลำดับของไบต์ที่เตรียมไว้มาส่งเป็นสัญญาณวิทยุออกไปยังเครื่องแม่ข่าย และเมื่อ CC1100 ส่งสัญญาณวิทยุเรียบร้อยแล้วไมโครคอนโทรเลอร์จะส่งสัญญาณไปยัง CC1100 เพื่อเปลี่ยนโหมดการทำงานของ CC1100 ให้เข้าสู่โหมดประหยัดพลังงานด้วยการเข้าสู่โหมดหลับ เพื่อลดการใช้พลังงานให้น้อยที่สุดดังจะได้แสดงรายละเอียดในหัวข้อต่อไป

3.1.8 การประหยัดพลังงาน

บอร์ด Arduino จะใช้พลังงานในโหมดปกติประมาณ 35 มิลลิแอมป์ (Arduino, 2014) รวมทั้งในบอร์ด Arduino บางรุ่นจะต้องใช้พลังงานประมาณ 30 มิลลิแอมป์สำหรับการสื่อสารระหว่างผานทางพอร์ท USB 868/915 (Texas Instruments, 2014) นอกจากนั้นแล้วชิป CC1100 ยังต้องการพลังงานอีกxxxมิลิแอมป์ระหว่างการส่งสัญญาณวิทยุ ซึ่งการใช้กระแสไฟขนาดนี้ถือว่าเป็นจำนวนมากหากบอร์ด Arduino นั้นใช้แหล่งพลังงานจากภายนอก โดยเฉพาะอย่างยิ่งจาก แบตเตอรี่ ดังนั้นหากเราไม่ได้ใช้งานบอร์ด เราจึงควรลดการใช้พลังงานของบอร์ด Arduino ให้เหลือน้อยที่สุดเท่าที่จะเป็นไปได้ เทคนิคโดยทั่วไปที่จะทำคือการเลือกปิดอุปกรณ์บางชิ้นบนไมโครคอนโทรเลอร์เมื่อไม่มีความจำเป็นที่จะต้องถูกเรียกใช้งาน

โดยหลักการแล้วบอร์ด Arduino จะทำงานแบบประหยัดพลังงานได้โดยการปิดการทำงานของอุปกรณ์บางอย่างที่ไม่จำเป็น เช่นเอดีซีและพอร์ตสื่อสารต่างๆ โดยที่บอร์ด Arduino มีโหมดการประหยัดพลังงานพื้นฐานให้ผู้ใช้เลือกใช้อยู่ทั้งหมดห้าโหมด ดังรายละเอียดในตารางที่ 3.7

ตารางที่ 3.7 การทำงานของบอร์ด Arduino ในโหมดประหยัดพลังงานแบบต่างๆ
ที่มา: Nawrath. 2014

Sleep mode	Active Clock Domains					Oscillators			Wake-up Sources							Software BOD Disble
Idle																
ADC Noise Reduction																
Power-down																
Power-save																
Standby																
Extended Standby																

เราสามารถสั่งให้ไมโครคอนโทรเลอร์เข้าสู่โหมดเดินเครื่องเปล่า ได้ด้วยการเรียกใช้ฟังก์ชั่น sleep() พร้อมกับผ่านค่า SLEEP_MODE_IDLEเป็นอาร์กิวเม้นต์ให้กับฟังก์ชั่น ซึ่งในโหมดนี้จะเป็นโหมดที่ประหยัดพลังงานน้อยที่สุดในบรรดาห้าโหมด สำหรับการทำงานในโหมด SLEEP_MODE_ADC นั้นไมโครคอนโทรเลอร์จะปิดการทำงานของเอดีซี และสำหรับการทำงานในโหมด SLEEP_MODE_PWR_DOWN นั้นไมโครคอนโทรเลอร์จะปิดการทำงานของโมดูลเอดีซีและบีโอดี (brown-out detection : BOD) รวมทั้งพอร์ตเชื่อมต่ออื่น ๆ ทั้งหมด ดังนั้นการทำงานโหมดนี้จึงเป็นโหมดการทำงานที่ประหยัดพลังงานได้มากที่สุด จากเหตุผลดังกล่าว งานวิจัยนี้จึงสั่งให้บอร์ด Arduino ประหยัดพลังงานด้วยการสลับการทำงานไปอยู่ทำงานในโหมด SLEEP_MODE_PWR_DOWN เมื่อไม่ได้ใช้งาน

อย่างไรก็ตามเมื่อเข้าสู่โหมดหลับแบบ SLEEP_MODE_PWR_DOWN แล้วไมโครคอนโทรเลอร์ก็จะอยู่ในสถานะหลับตลอดไปจนกว่าจะมีการขัดจังหวะ(interrupt) เกิดขึ้น นั่นหมายความว่าหากไม่มีสัญญาณขัดจังหวะส่งไปหาบอร์ด หน่วยประมวลผลแล้ว ไมโครคอนโทรเลอร์นั้นก็จะหลับต่อไปจนกว่าผู้ใช้จะทำการรีเซตอุปกรณ์ด้วยตนเอง

วิธีการปลุกให้บอร์ด Arduino ด้วยการขัดจังหวะนั้นทำได้หลายแนวทาง แนวทางแรกคือการตรวจสอบการขัดจังหวัดภายนอก (external interrupt) ของบอร์ด Arduino ซึ่งมักจะทำโดยใช้ตัวจับเวลาภายนอกเป็นตัวส่งสัญญาณเข้ามายังขาดิจิทัลอินพุตของบอร์ด Arduinoเพื่อให้มีการขัดจังหวะเกิดขึ้น

ซึ่งบอร์ด Arduino นั้นออกแบบมาให้ผู้ใช้สามารถส่งสัญญาณขัดจังหวะเข้ามาได้โดยใช้ขาดิจิทัลขาที่สองและสามแนวทางที่สองที่จะขัดจังหวะการหลับได้คือการตรวจสอบสัญญาณการสื่อสารแบบอนุกรมจากพอร์ต USART (universal synchronous-asynchronous receiver/transmitter)เช่นพอร์ตอนุกรมหรือ USB อย่างไรก็ตามการจะปลุกด้วยวิธีนี้บอร์ด Arduino จะต้องหลับในโหมด POWER_MODE_IDLE เท่านั้น ทั้งนี้เพราะการหลับในโหมดนี้จะไม่ปิดการทำงานของ USART (แต่โหมดอื่น ๆ จะปิด)อีกแนวทางหนึ่งคือการใช้ตัวจับเวลาภายใน (internal timer) ซึ่งการปลุกด้วยวิธีนี้ก็สามารถปลุกบอร์ด Arduino ที่หลับในโหมด SLEEP_MODE_PWR_DOWN ได้ดีเช่นเดียวกัน

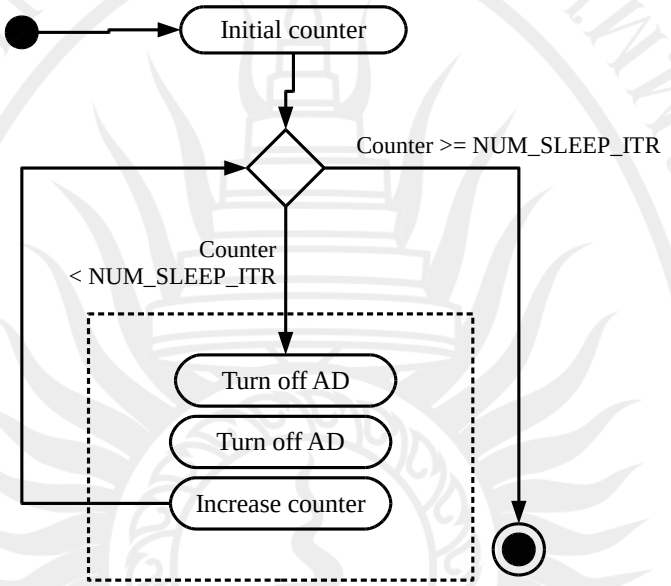
และแนวทางสุดท้ายคือการสั่งให้บอร์ดหลับและตื่นโดยใช้ตัวจับเวลาควบคุมการทำงานหรือดับเบิลยูดีที (watchdog timer : WDT) การปลุกด้วยวิธีนี้ก็สามารถปลุกบอร์ด Arduino ที่หลับในโหมด SLEEP_MODE_PWR_DOWNได้เช่นเดียวกัน ตารางที่ 3.8แสดงรายละเอียดเกี่ยวกับการกำหนดค่าของดับเบิลยูดีที

ตารางที่ 3.8 ช่วงเวลาของดับเบิลยูดีที
ที่มา: Nawrath. 2014

WDP3	WDP2	WDP1	WDP0	No. of WDT Oscillator Cycles	Typical Time-out at V_{cc} = 5V
0	0	0	0	2K (2048) cycles	16 ms
0	0	0	1	4K (4096) cycles	32ms
0	0	1	0	8K (8192) cycles	64ms
0	0	1	1	16K (16384) cycles	0.125 s
0	1	0	0	32K (32668) cycles	0.25 s
0	1	0	1	64K (65536) cycles	o.5 s
0	1	1	0	128K (131072) cycles	1.0 s
0	1	1	1	256K (262144) cycles	2.0 s
1	0	0	0	512K (524288) cycles	4.0 s
1	0	0	1	1024K (1048576) cycles	8.0 s

เมื่อพิจารณาเงื่อนไขต่างๆร่วมกับข้อมูล จากตารางนี้ งานวิจัยนี้ใช้จึงกำหนดใช้ดับเบิลยูดีทีในการสั่งให้บอร์ด Arduino ประหยัดพลังงานเป็นเวลานานที่สุดเท่าที่ดับเบิลยูดีทีจะทำได้ นั่นคือกำหนดให้บอร์ด Arduino ปิดการทำงานของโมดูลเอดีซีและโมดูลบีโอดี หลังจากนั้นแล้วให้บอร์ดหยุดการทำงานทั้งหมดเป็นเวลาแปดวินาที อย่างไรก็ตามเมื่อไมโครคอนโทรเลอร์ตื่นจากโหมดหลับแล้วก็จะเข้าสู่โหมดการทำงานปกติที่ไม่ประหยัดพลังงานเช่นเดิม ดังนั้นเพื่อให้ไมโครคอนโทรเลอร์อยู่ในสภาวะหลับอยู่นาน

มากกว่าแปดวินาที ทุกครั้งที่ไมโครคอนโทรเลอร์ตื่นขึ้นมา ผู้วิจัยจะใช้เทคนิคการเขียนโปรแกรมวนรอบเข้ามาสั่งให้ไมโครคอนโทรเลอร์กลับเข้าสู่โหมดหลับอย่างต่อเนื่องจนกว่าเวลาจะผ่านไปนานเท่าที่ได้กำหนดไว้ ภาพที่ 3.8 แสดงตัวอย่างผังการทำงานของการควบคุมให้ไมโครคอนโทรเลอร์อยู่ในโหมดหลับเป็นเวลา 10 นาที (ในที่นี้กำหนดให้ NUM_SLEEP_ITR มีค่าเท่ากับ 75)



ภาพที่ 3.8 ผังงานการควบคุมการหลับของบอร์ด Arduino ด้วยดับเบิลยูดีที

3.2 อุปกรณ์ฝั่งแม่ข่าย

อุปกรณ์ฝั่งแม่ข่ายในกลุ่มนี้จะแบ่งออกเป็นสองส่วน ส่วนแรกคือส่วนที่เกี่ยวข้องกับการจัดการข้อมูลที่ส่งผ่านมาทางสัญญาณวิทยุ และส่วนที่สองคืออุปกรณ์ที่เกี่ยวข้องกับการควบคุมการเปิดปิดน้ำตามความต้องการของผู้ใช้ และรายละเอียดการออกแบบการทำงานของอุปกรณ์ทั้งสองส่วนมีดังต่อไปนี้

3.2.1 ส่วนการรับสัญญาณวิทยุ

ที่เครื่องคอมพิวเตอร์แม่ข่ายหลักจะมีตัวรับส่งวิทยุเชื่อมต่ออยู่ (ใช้บอร์ด Arduino และชิป CC1100 เช่นเดียวกับระบบส่วนลูกข่าย) โดยอุปกรณ์นี้จะคอยฟังสัญญาณวิทยุที่ส่งมาจากบอร์ด Arduino ของเครื่องลูกข่าย หากพบว่ามีสัญญาณในย่านความถี่ที่กำหนดไว้ปรากฏเข้ามา ชิป CC1100 ก็จะรับสัญญาณนั้นแล้วถอดรหัสสัญญาณวิทยุที่ส่งมาให้อยู่ในรูปของแถวลำดับของไบต์ หลังจากนั้นไมโครคอนโทรเลอร์จะทำการตรวจสอบความสมเหตุสมผล (verify) ว่าข้อมูลที่รับได้นั้นเป็นข้อมูลที่ต้องการใช่หรือไม่ ซึ่งจะทำโดยการตรวจสอบว่าข้อมูลส่วนหัวของข้อมูลที่รับได้นั้นตรงกับค่าที่กำหนดหรือไม่ หากพบว่าข้อมูลมีความสมเหตุสมผลและมีค่าอยู่ในช่วงที่ควรจะเป็นแล้วก็จะทำการบันทึกข้อมูลเก็บไว้เป็นฐานข้อมูลอย่างไรก็ตามบอร์ด Arduino UNO เป็นไมโครคอนโทรเลอร์ที่ไม่มีฟังก์ชั่นการจัดการไฟล์และฐานข้อมูล นอกจากนั้นแล้วก็ยังไม่มีฟังก์ชั่นด้านวันที่และเวลาที่เหมาะสม

ดังนั้นผู้วิจัยจึงกำหนดให้บอร์ด Arduino ส่งข้อมูลต่าง ๆ ที่รับได้ต่อไปให้โปรแกรมประยุกต์อีกโปรแกรมหนึ่งเพื่อการบันทึกข้อมูลดังกล่าวโปรแกรมประยุกต์ดังกล่าวเป็นโปรแกรมที่ผู้วิจัยพัฒนาด้วยภาษา Processing ซึ่งผู้วิจัยออกแบบให้รันบนระบบปฏิบัติการวินโดวส์ โดยโปรแกรมนี้จะใช้เพื่อทำหน้าที่ดักฟังข้อมูลที่ส่งมาจากบอร์ด Arduino ทางพอร์ตอนุกรม หากพบว่ามีข้อมูลส่งมา โปรแกรมนี้ก็จะสกัดสายอักขระที่ได้รับให้พร้อมใช้งาน อย่างไรก็ตามข้อมูลที่ส่งมาจากบอร์ด Aruino นั้นไม่มีข้อมูลเกี่ยวกับวันที่และเวลาที่เซนเซอร์ได้ทำการวัดข้อมูล ดังนั้นผู้วิจัยจึงกำหนดว่าเมื่อ Processing ได้รับข้อมูลแล้ว ก็จะให้เพิ่มข้อมูลวันที่และเวลาที่ได้รับข้อมูลเพิ่มเติม โดยใช้ฟังก์ชั่นด้านวันที่และเวลาของคอมพิวเตอร์แม่ข่ายที่โปรแกรม Processing ทำงานอยู่ ทั้งนี้แม้ว่าอาจจะมีความเหลื่อมระหว่างเวลาที่ใช้ในการสื่อสารจากพื้นที่ศึกษามายังเครื่องแม่ข่าย แต่เนื่องจากการสื่อสารดังกล่าวเป็นการสื่อสารด้วยคลื่นวิทยุและระยะทางระหว่างเครื่องส่งและเครื่องรับมีระยะทางสั้น (ไม่เกิน 100 เมตร) ดังนั้นความคลาดเคลื่อนด้านเวลาระหว่างที่วัดค่าได้กับเวลาที่เครื่องแม่ข่ายได้รับข้อมูลจึงมีค่าน้อยมาก ดังนั้นงานวิจัยนี้จึงไม่สนใจความคลาดเคลื่อนเรื่องเวลาดังกล่าว

3.2.2 การออกแบบฐานข้อมูล

เมื่อโปรแกรมที่พัฒนาด้วย Processing ได้สกัดและเตรียมข้อมูลต่าง ๆ เรียบร้อยแล้วก็จะส่งข้อมูลเหล่านี้ไปบันทึกที่เครื่องคอมพิวเตอร์แม่ข่ายเครื่องที่สองผ่านทางอินเทอร์เน็ต ที่เครื่องแม่ข่ายเครื่องที่สองนี้จะมีสคริปต์ที่ผู้วิจัยพัฒนาด้วยภาษา PHP เพื่อทำหน้าที่รับข้อมูลการตรวจวัดที่ส่งเข้ามา สคริปต์นี้จะสกัดข้อมูลที่รับได้ให้อยู่ในรูปพร้อมที่จะใช้งาน ตรวจสอบข้อมูลที่ได้รับ หลังจากนั้นจึงส่งข้อมูลที่ตรวจสอบความสมเหตุสมผลแล้วไปเก็บลงฐานข้อมูลต่อไป

โดยสรุปแล้วข้อมูลที่จะทำการบันทึกลงในฐานข้อมูลนั้นจะประกอบด้วย หมายเลขเครื่องตรวจวัด, วันที่, เวลา, อุณหภูมิของอากาศ, ความชื้นสัมพัทธ์ของอากาศ, อุณหภูมิของดิน, ความชื้นสัมพัทธ์ของดิน และความเข้มแสงแวดล้อมโดยงานวิจัยนี้ใช้ได้ระบบจัดการฐานข้อมูล MySQL เป็นหลักและมีรายละเอียดของการออกแบบฐานข้อมูลดังแสดงในตารางที่ 3.9

ตารางที่ 3.9 การออกแบบฐานข้อมูลสำหรับบันทึกข้อมูลที่วัดได้

ชื่อฟิลด์	ความหมาย	ชนิดข้อมูล	พิสัยข้อมูล	ค่าโดยปริยาย	หมายเหตุ
SID	รหัสข้อมูล	INT			PK, AUTO
SENSOR_ID	รหัสอุปกรณ์	INT			
SOIL_TEM	อุณหภูมิดิน	FLOAT		0 - 100	
SOIL_RH	ความชื้นสัมพัทธ์ดิน	FLOAT		0 - 100	
AIR_TEM	อุณหภูมิอากาศ	FLOAT		0 - 100	
AIR_RH	ความชื้นสัมพัทธ์อากาศ	FLOAT		0 - 100	
AMB_LIGHT	ความเข้มแสงแวดล้อม	INT		0 - 255	
S_DATE	วันที่ที่วัดข้อมูล	char			
S_TIME	เวลาที่วัดข้อมูล	char			

3.2.3 การควบคุมการเปิดปิดโซเลนอยด์

ที่เครื่องคอมพิวเตอร์แม่ข่ายเครื่องแรก จะมีโซเลนอยด์เชื่อมต่ออยู่กับบอร์ด Arduino โดยที่โซเลนอยด์ตัวนี้จะทำหน้าที่ควบคุมการเปิดและปิดวาล์วน้ำตามความต้องการของผู้ใช้ โดยงานวิจัยนี้ได้เลือกใช้โซเลนอยด์รุ่น AQT15SP ซึ่งเป็นโซเลนอยด์แบบสองพอร์ต ใช้แรงดันไฟ 12 โวลต์ มีท่อทางเข้า (inlet) เป็นพลาสติกขนาด 1/2 นิ้ว และท่อทางออก(outlet)ขนาด 12 มิลลิเมตร โดยจะทำงานได้เมื่อมีแรงดันน้ำอย่างน้อย 0.02 เมกะปาสกาล ซึ่งสถานะปกติของโซเลนอยด์นี้จะเป็นแบบปกติปิด (normally closed : N.O.) หมายถึงจะมีสถานะปกติเมื่อไม่มีสัญญาณไฟฟ้าเป็นปิดการทำงานซึ่งจะทำให้น้ำไหลผ่านไปไม่ได้ ภาพที่ User Field CHAPTER_NO = 3. Illustrationแสดงตัวอย่างของโซเลนอยด์ที่ใช้ในการวิจัยนี้

ภาพที่ 3.9 โซเลนอยด์ AQT15SP

ในการเชื่อมต่อกับบอร์ด Arduino นั้น ผู้วิจัยได้ออกแบบให้ใช้ทรานซิสเตอร์ร่วมกับไดโอดเพื่อจัดการกับกระแสไฟฟ้า ที่จะผ่านไปยังโซเลนอยด์ โดยในกรณีของงานวิจัยนี้จะควบคุมการเปิดและปิด โซเลนอยด์โดยใช้การส่งสัญญาณดิจิทัลจากไมโครคอนโทรเลอร์ไปยังโซเลนอยด์ โดยเมื่อเครื่องแม่ข่ายพบว่าผู้ใช้มีการส่งสัญญาณความต้องการเปิดน้ำมายังเครื่องแม่ข่าย บอร์ด Arduino ก็จะส่งสัญญาณ HIGH ออกไปยังขาดิจิทัลที่ได้กำหนดไว้เพื่อเปิดการทำงานของวาล์วของโซเลนอยด์และจะทำให้น้ำไหลผ่านได้ และหากพบว่าผู้ใช้ต้องการปิดน้ำ บอร์ด Arduino ก็จะส่งสัญญาณ LOW ออกไปทางขาดิจิทัลเพื่อปิดวาล์วของโซเลนอยด์ซึ่งจะทำให้น้ำไหลผ่านไม่ได้

งานวิจัยนี้ออกแบบให้ผู้ใช้สื่อสารกับบอร์ด Arduino เพื่อควบคุมการทำงานของโซเลนอยด์ผ่านทางอุปกรณ์เคลื่อนที่ ซึ่งในกรณีนี้คือ สมาร์ทโฟน และเพื่อให้เกิดความยืดหยุ่นโดยไม่สนใจระบบปฏิบัติการของอุปกรณ์เคลื่อนที่ ผู้วิจัยจึงได้พัฒนาโปรแกรมประยุกต์ที่ทำงานบนเว็บเพื่อที่จะให้ผู้ใช้ป้อนความต้องการเปิดปิดน้ำดังกล่าวได้โดยไม่ขึ้นกับระบบปฏิบัติการในภาพรวมนั้นโปรแกรมนี้จะมีปุ่มอินพุตสองปุ่มให้ผู้ใช้ป้อนความต้องการเปิดหรือปิดน้ำและเมื่อผู้ใช้กดปุ่มแล้วข้อมูลความต้องการนี้จะถูกบันทึกลงบนเครื่องแม่ข่ายเครื่องที่สอง (เครื่องเดียวกับที่ใช้บันทึกข้อมูลลงฐานข้อมูล)

ที่เครื่องแม่ข่ายเครื่องที่สอง จะมีโปรแกรมที่พัฒนาด้วยภาษา PHP ซึ่งจะทำหน้าที่บันทึกสถานะของโซเลนอยด์ไว้ และเนื่องจากข้อมูลสถานะมีขนาดเพียงไม่กี่ไบต์และไม่ได้มีความถี่ในการใช้งานสูง ดังนั้นผู้วิจัยจึงกำหนดให้สคริปต์ PHP จะบันทึกข้อมูลนี้เป็นแฟ้มข้อความ(text file)เพื่อความสะดวกในการอ่านด้วย Processing โดยผู้วิจัยกำหนดให้ไฟล์นี้บันทึกเลข1 หากผู้ใช้ต้องการเปิดวาล์วน้ำ และให้บันทึกเลข0 แทนความผู้ใช้ต้องการปิดวาล์วน้ำ

อย่างไรก็ตาม นั้นเป็นไมโครคอนโทรเลอร์ที่ไม่มีความสามารถทางด้านการจัดการไฟล์ ดังนั้นผู้วิจัยจึงได้พัฒนาโปรแกรมประยุกต์ด้วยภาษา Processing ขึ้นมาอีกหนึ่งโปรแกรม โดยโปรแกรมนี้จะเป็นตัวกลางในการสื่อสารระหว่างผู้ใช้และบอร์ด Arduino โดยโปรแกรมProcessing โปรแกรมนี้จะอ่านข้อมูลสถานะผ่านเครือข่ายอินเทอร์เน็ตจากเครื่องแม่ข่ายว่าสถานะของวาล์วนั้นเป็นค่าอะไร หากพบว่ามีค่าเป็น 1 แล้ว โปรแกรมนี้ก็จะส่งข้อมูลไปยังไมโครคอนโทรเลอร์ผ่านทางพอร์ตอนุกรม เพื่อให้ไมโครคอนโทรเลอร์รู้ว่ามีการเปลี่ยนแปลงสถานะ และเมื่อไมโครได้รับสัญญาณแล้วจะส่งสัญญาณไปยังโซเลนอยด์ตามรายละเอียดที่ได้กล่าวมาแล้วในหัวข้อข้างต้น