



รายงานวิจัยฉบับสมบูรณ์

โครงการ

การพัฒนาสถานีตรวจวัดอากาศแบบไร้สายและระบบสารสนเทศ
เพื่อการเกษตรแบบแม่นยำ

Development of wireless weather station for precision agriculture

โดย

เกริก ปั่นเหล่งเพ็ชร

31 สิงหาคม 2560

สัญญาเลขที่ SRI5920205

รายงานวิจัยฉบับสมบูรณ์

โครงการ

การพัฒนาสถานีตรวจวัดอากาศแบบไร้สายและ
ระบบสารสนเทศเพื่อการเกษตรแบบแม่นยำ

Development of wireless weather station
for precision agriculture

คณะผู้วิจัย

เกริก ปั่นเหน่งเพชร

สนับสนุนโดยสำนักงานกองทุนสนับสนุนการวิจัย (สกว.)
(ความเห็นในรายงานนี้เป็นของผู้วิจัย สกว.ไม่จำเป็นต้องเห็นด้วยเสมอไป)

สารบัญ

	หน้า
Executive Summary	iii
Abstract	iv
บทคัดย่อ	v
เนื้อหาทางนวิจัย	1
บทนำ	1
วัตถุประสงค์หลักของโครงการ	1
การดำเนินงานในระยะที่ 1	2
การพัฒนา 3 จี โมดูล	2
การพัฒนาเครื่องบันทึกข้อมูล	2
การพัฒนาแหล่งจ่ายไฟ	4
การดำเนินงานในระยะที่ 2	5
การสร้างเซ็นเซอร์ตรวจสอบสภาพอากาศ	5
การเทียบค่าของเซ็นเซอร์	7
การประกอบสถานีตรวจวัดสภาพอากาศ	7
การตรวจสอบการทำงานและความแม่นยำของสถานีตรวจวัดสภาพอากาศเบื้องต้น	9
การติดตั้งและทดสอบภาคสนาม	10
การดำเนินงานในระยะที่ 3	11
การทดสอบในช่วงฤดูฝนต่อเนื่องจากระยะที่ 2	11
ปัญหาที่พบและได้แก้ไข	11
ข้อจำกัด	11
สรุปผลการดำเนินการ	13
เอกสารอ้างอิง	14
ภาคผนวกที่ 1 ชุดคำสั่งระบบปฏิบัติการของสถานีตรวจวัดสภาพอากาศ	16
ภาคผนวกที่ 2 ขั้นตอนการวิเคราะห์และแก้ไขปัญหา	27
ภาคผนวกที่ 3 ยกร่าง manuscript เพื่อการตีพิมพ์ในวารสารนานาชาติ	43
ตารางเปรียบเทียบวัตถุประสงค์ กิจกรรมที่วางแผนไว้ และกิจกรรมที่ดำเนินการมาและผลที่ได้รับ	52

Executive Summary

Needs to maximize input efficiency and crop yield while minimize impact on environment have driven food production system toward information based precision agriculture, in which near real time weather information is one of the fundamental components. Therefore, automatic weather station that was previously developed and funded by TRF (RDG53O0009 and RDG 5650128) was extended to enable transmission of near real time weather data from the field via 3G-internet network to a web server.

3G module was designed together with the modifications of electronic circuit and software of the previously developed datalogger so that the station could connect to the internet via cellular service provider access point and send weather data to a web server. Twenty wireless weather stations were then built, and thirteen stations were installed in the field: nine in orange groves in Chai Prakan, Fang and Mae Ai district, ChaingMai province: one at Mae Hia Agricultural Research Center, Faculty of Agriculture, ChiangMai University: one in Mueang district, Phayao Province: one in Suan Kahakarkaset, Lat Lum Kaeo distric, Phatum thani Province: and one in vineyard, Muak Lek district, Saraburi province. The remaining stations are to be installed to support other projects after the end of this project.

Testing under field conditions from April to September showed that the wireless weather stations developed in this project consistently sent hourly weather data from the field to web server correctly. To ensure reliable data transmission, especially during heavy rain, the station should be installed in area where 3G signal strength is better than < -90 dBm (-100 dBm or lower is generally considered as no signal).

Abstract

Needs to maximize input efficiency and yield while minimize impact on environment have driven food production system toward information based precision agriculture, in which near real time weather information is one of the fundamental components. Therefore, automatic weather station that was previously developed and funded by TRF (RDG53O0009 and RDG 5650128) was upgraded to allow transmission of near real time weather data from the field via 3G-internet network to a web server, where other projects within the frame work of precision agriculture research development and application can freely access.

In the first 6 month, 3G module was designed together with the modifications of electronic circuit and software of the previously developed datalogger so that the station could connect to the internet via cellular service provider access point and send weather data to a web server.

In the second 6 months, 20 wireless weather stations were built, comprising of pyranometer, air temperature and humidity sensor, anemometer, rain gauge, datalogger and 3G module. Nine stations were installed and tested in orange groves in Chai Prakan, Fang and Mae Ai district, ChiangMai province, and one station at Mae Hia Agricultural Research Center, Faculty of Agriculture, ChiangMai University. The weather stations sent data through 3G cellular network-Internet to the web server of the Center for Agricultural Resource, ChiangMai Univeristy, without any loss of data. Precision of the measurements of solar radiation, air temperature and humidity, and rainfall was high. For wind speed, small but noticeable variation was observed when wind speed was less than 0.45 m s^{-1} (light air).

In the third 6 months, field testing was conducted further into the rainy season, and three more stations were installed in the field: one in Mueang district, Phayao Province: one in Suan Kahakarkaset, Lat Lum Kaeo distric, Phatum thani Province: and one in vineyard, Muak Lek district, Saraburi province. During this period it was found that some of the connectors that connected sensors to the datalogger became creep/deformed, and generated noise in the signal sent from sensor to datalogger. The problem was solved by replacing the connectors with pin-socket type. It was also observed that data transmission of two weather stations installed in area where 3G signal strength was marginal ($< -95 \text{ dBm}$) was not stable, especially during heavy rain, and they were consequently removed . Hence -90 dBm is considered as the minimal RSSI for the weather station.

Key words: wireless weather station, precision agriculture, food security.

บทคัดย่อ

ความจำเป็นที่จะต้องเพิ่มผลผลิตพืชและประสิทธิภาพการใช้ทรัพยากรธรรมชาติให้สูง แต่ในขณะเดียวกันต้องลดผลกระทบที่มีต่อสภาพแวดล้อม ทำให้ระบบการเกษตรต้องใช้การจัดการที่แม่นยำ เฉพาะเจาะจงกับคุณสมบัติของแต่ละพื้นที่ ระบบการเกษตรดังกล่าวจำเป็นต้องใช้ข้อมูลที่มีรายละเอียดมากขึ้น และข้อมูลสภาพอากาศเป็นข้อมูลที่สำคัญประการหนึ่ง ดังนั้นโครงการนี้จึงนำสถานีตรวจวัดอากาศอัตโนมัติที่ได้พัฒนาขึ้นภายใต้การสนับสนุนของ สกว. (RDG5300009 และ RDG 5650128) มาเพิ่มคุณสมบัติให้สามารถส่งข้อมูลสภาพอากาศจากแปลง ผ่านเครือข่าย 3 จี และอินเทอร์เน็ต ไปยังเว็บแอปพลิเคชัน เพื่อให้โครงการอื่นในกรอบงานงานวิจัยพัฒนาและนำใช้เกษตรแม่นยำ ที่ได้รับมอบหมายจาก ฝ่าย ๒ สกว. นำไปใช้ประโยชน์

การดำเนินการในระยะที่ 1 ได้ออกแบบและสร้างแผงวงจรของ 3 จี โมดูล และนำเครื่องบันทึกข้อมูลที่พัฒนาไว้แล้วมาปรับเปลี่ยนวงจรอิเล็กทรอนิกส์และระบบปฏิบัติการให้สามารถเชื่อมต่อกับระบบอินเทอร์เน็ต ผ่านจุดเชื่อมต่อไร้สายของผู้ให้บริการเครือข่าย และส่งข้อมูลสภาพอากาศไปเก็บไว้ที่เว็บแอปพลิเคชัน

การดำเนินการในระยะที่ 2 ได้สร้างสถานีตรวจวัดอากาศ จำนวน 20 ชุด ซึ่งประกอบด้วยเซ็นเซอร์ วัดพลังแสงอาทิตย์ อุณหภูมิ ความชื้นอากาศ ความเร็วลม และปริมาณฝน เครื่องบันทึกข้อมูล 3 จี โมดูล และชุดให้พลังงานไฟฟ้า จากนั้นได้ติดตั้งสถานีดังกล่าวในสวนส้มในอำเภอไชยปราการ ผาง และแม่ฮาด จังหวัดเชียงใหม่ จำนวน 9 ชุด และ 1 ชุดที่ศูนย์วิจัยแม่เหียะ คณะเกษตรศาสตร์ มหาวิทยาลัย เชียงใหม่ สถานีตรวจวัดอากาศอัตโนมัติสามารถส่งข้อมูลสภาพอากาศไปยังเว็บเซิร์ฟเวอร์ ของศูนย์วิจัยระบบทรัพยากรเกษตร มหาวิทยาลัยเชียงใหม่ โดยไม่มีการสูญเสียข้อมูล ความแม่นยำของการวัดพลังแสงอาทิตย์ อุณหภูมิ ความชื้นอากาศ มีความแม่นยำสูง แต่สำหรับความเร็วลมอาจมีความแตกต่างพอสมควรได้ระหว่างเครื่องภายใต้สภาพลมสงบ (<0.45 เมตรต่อวินาที)

การดำเนินการในระยะที่ 3 เป็นการทดสอบในช่วงฤดูฝนต่อเนื่องจากระยะที่ 2 และได้ติดตั้งสถานีตรวจวัดสภาพอากาศเพิ่มเติมอีกในสามพื้นที่คือ ที่ ต.บ้านใหม่ อ.เมือง จ.พะเยา สวนเคหะการเกษตร อ.ลาดหลุมแก้ว จ.ปทุมธานี และ ในไร่ร่องน อ.มวกเหล็ก จ.สระบุรี การทดสอบในระยะนี้พบว่าปลั๊กที่ใช้เชื่อมต่อเครื่องบันทึกข้อมูลกับเซ็นเซอร์บางตัวเกิดการคายตัวทำให้ค่าที่เครื่องบันทึกข้อมูลได้รับจากเซ็นเซอร์คลาดเคลื่อนจึงได้ทำการแก้ไขโดยเปลี่ยนไปใช้ปลั๊กแบบ pin-socket นอกจากนี้ยังพบว่าเครื่อง 2 ชุดที่ติดตั้งในพื้นที่ที่มีสัญญาณ 3จีต่ำ (<-95 dBm) การส่งข้อมูลจะไม่เสถียร จึงได้นำเครื่อง 2 ชุดดังกล่าวออกจากพื้นที่ดังกล่าว ดังนั้นจึงถือว่า -90 dBm เป็นระดับความแรงขั้นต่ำของสัญญาณ 3จี สำหรับสถานีตรวจวัดสภาพอากาศแบบไร้สายนี้

คำสำคัญ: สถานีตรวจวัดสภาพอากาศแบบไร้สาย การเกษตรแบบแม่นยำ ความมั่นคงด้านอาหาร

เนื้อหางานการวิจัย

บทนำ

การขยายตัวของประชากร ที่อยู่อาศัย และอุตสาหกรรมเกิดขึ้นอย่างรวดเร็ว ทำให้พื้นที่ปลูกพืชและปัจจัยอื่น ๆ ที่ต้องใช้ในการผลิตถูกจำกัดมากขึ้น ประกอบกับความแปรปรวนของสภาพอากาศที่รุนแรงขึ้นจากภาวะโลกร้อน ได้ผลักดันให้ระบบการเกษตรต้องใช้การจัดการที่แม่นยำและมีความเฉพาะเจาะจงกับพื้นที่มากขึ้น (site specific management / precision agriculture) ทั้งนี้เพื่อให้สามารถใช้ทรัพยากรในแต่ละพื้นที่ที่แตกต่างกันได้อย่างมีประสิทธิภาพ

ระบบการเกษตรแบบแม่นยำจำเป็นต้องใช้ข้อมูลสภาพอากาศที่มีรายละเอียดสูง ทั้งในแง่ของเวลาและสถานที่ และเป็นข้อมูลแบบใกล้เคียงปัจจุบัน (near real time) ปัจจุบันได้มีการนำเทคโนโลยีแบบไร้สายมาใช้ในการส่งข้อมูลสภาพแวดล้อมจากเซ็นเซอร์ ที่วัดในแปลงปลูกไปยังเซิร์ฟเวอร์ เพื่อให้สามารถวิเคราะห์ข้อมูลได้แบบใกล้เคียงปัจจุบัน (near real time) ระบบไร้สายที่นำมาใช้นั้นมีทั้งการส่งข้อมูลระยะไกลผ่านระบบเครือข่ายเซลลูลาร์-อินเทอร์เน็ต (Liqiang et al 2011; Shining et al 2011; Kassim et al. 2014; Chaudhary et al, 2011; Sakthipriya. 2014; และBaggio, 2015) และระยะใกล้ผ่านระบบ zigbee/bluetooth/wifi เป็นต้น (Rehman et al, 2014; Camilli et al, 2007; Keshtgary and Deljoo, 2012; Wang et al, 2006; และ Garcia-Sanchez et al. 2011) ดังนั้น โครงการนี้จึงนำสถานีตรวจวัดอากาศอัตโนมัติที่ได้พัฒนาขึ้นภายใต้การสนับสนุนของ สกว. RDG53O0009 (เกริก 2555) และได้ผ่านการทดสอบมาตรฐานเบื้องต้น สกว. RDG 5650128 (เกริก และคณะ 2557) มาเพิ่มคุณสมบัติให้สามารถส่งข้อมูลสภาพอากาศที่วัดในแปลงผ่านเครือข่าย 3G และอินเทอร์เน็ต ไปเก็บไว้ยัง เว็บเซิร์ฟเวอร์ เพื่อใช้ในระบบการเกษตรแบบแม่นยำ

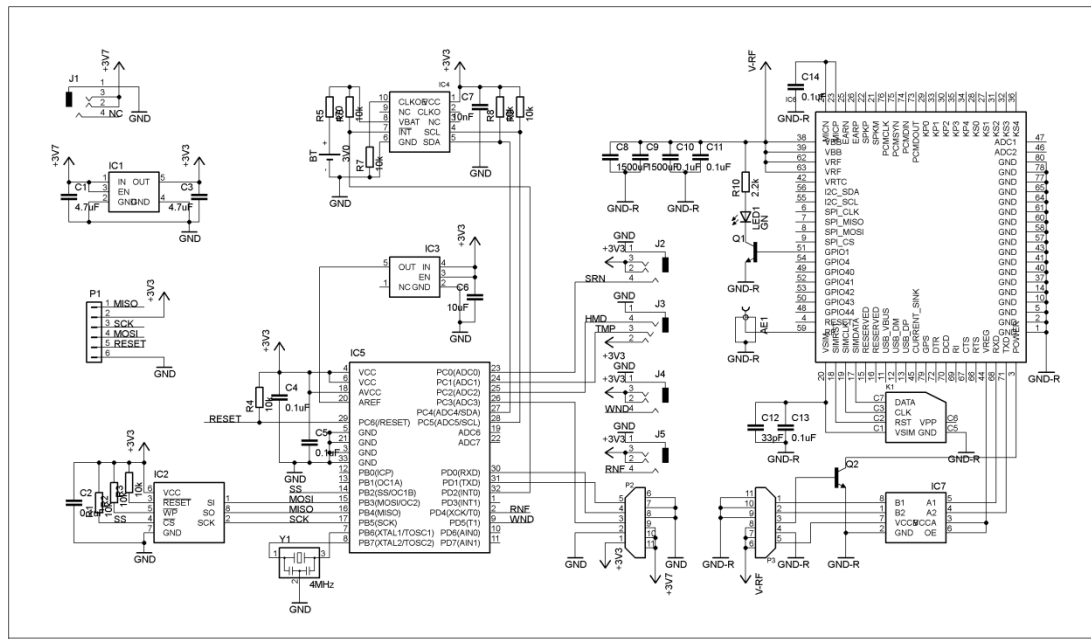
วัตถุประสงค์ของการวิจัย

พัฒนาสถานีตรวจวัดอากาศแบบไร้สายที่สามารถส่งข้อมูลสภาพอากาศ ณ เวลาใกล้เคียงปัจจุบันจากภาคสนาม ผ่านเครือข่าย 3 จี และอินเทอร์เน็ต ไปเก็บไว้ยังเว็บเซิร์ฟเวอร์ ของศูนย์วิจัยระบบทรัพยากรเกษตร คณะเกษตรศาสตร์ มหาวิทยาลัยเชียงใหม่ เพื่อให้โครงการอื่นซึ่งอยู่ในกรอบงานการเกษตรแบบแม่นยำ ที่ได้รับมอบหมายจาก ฝ่าย ๒ สกว. นำไปใช้ประโยชน์

การดำเนินงานในระยะที่ 1

การพัฒนา 3 จี โมดูล

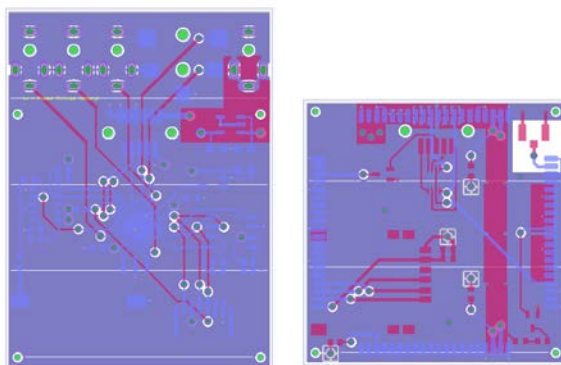
ได้ออกแบบและสร้างแผงวงจรของ 3 จี โมดูลขึ้น ซึ่งมีองค์ประกอบหลักคือ 3 จี SoC ที่ทำงานที่ความถี่ 2.1 GHz ช่องใส่ไมโครซิมการ์ด และตัวแปลงแรงดันไฟฟ้า เพื่อเชื่อมต่อกับเครื่องบันทึกข้อมูลผ่านช่องสื่อสาร USART ดังแสดงในภาพที่ 1



ภาพที่ 1 แผนผังวงจร 3 จี โมดูลที่เชื่อมต่อกับเครื่องบันทึกข้อมูลผ่านช่องสื่อสาร USART

การพัฒนาเครื่องบันทึกข้อมูล

ได้ปรับเปลี่ยนวงจรอิเล็กทรอนิกส์ของเครื่องบันทึกข้อมูลที่ได้พัฒนาไว้แล้ว (เกริก 2555) เพื่อให้สามารถเชื่อมต่อกับ 3 จี โมดูล ดังแสดงในรูปที่ 2 พร้อมกับปรับระบบปฏิบัติการของเครื่องบันทึกข้อมูล ที่เขียนโดยใช้ภาษา C เพื่อสั่งงานให้ 3 จี โมดูล ทำการเชื่อมต่อกับ access point ของผู้ให้บริการ 3 จี จากนั้นเชื่อมต่อกับเว็บเซิร์ฟเวอร์ โดยใช้ โพรโทคอล TCP/IP ตาม เลขที่อยู่ IP หรือ ชื่อโดเมน ที่ระบุไว้ และส่งข้อมูลในรูปแบบของ http ไปยัง URL ที่กำหนดไว้ ดังแสดงในรายการที่ 1



ภาพที่ 2 ลายพิมพ์วงจรของเครื่องบันทึกข้อมูล และ 3จี โมดูล ภาพขนาดใกล้เคียงของจริง

รายการที่ 1 ชุดคำสั่งที่ใช้สั่ง 3 จี โมดูล ให้เชื่อมต่อและส่งข้อมูลผ่านเครือข่าย 3 จี-อินเทอร์เน็ต ไปยัง URL ที่กำหนดไว้ และชุดคำสั่งทั้งหมดได้รายงานไว้ในภาคผนวกที่ 1

```
#include <avr/io.h>
#include <string.h>
#include "uart.h"
#include "udfm.h"
#include "ulog.h"
#include "wtcp.h"

static unsigned char net[] = "ATE0;+CREG?\r\n";
static unsigned char opn[] =
"AT+CGSOCKCONT=1,\"IP\", \"APN\";+NETOPEN=\"TCP\",80;+TCPCONNECT=\"HOST"
 "\",80\r\n";

static unsigned char cls[] = "AT+NETCLOSE\r\n";
static unsigned char pof[] = "AT+CPOF\r\n";

static unsigned char ok [] = "OK";
static unsigned char reg[] = "0,1";
static unsigned char prm[] = ">";
static unsigned char rcv[] = "OK";

static unsigned char ht0[] = "AT+TCPWRITE= \r\n PUT
/USERVER/"STID"/YYMM.TXT HTTP/1.1\r\nHost: "HOST"\r\nContent-Length:
\r\n\r\n";

unsigned char tcp(unsigned char op)
{
    unsigned char htx[1000] = {0};
    unsigned int ln0, lnx;

    PORTC = 0xFE;
    qsec(2);
    PORTC = 0xF6;
    qsec(20);
    sendcmd(net, reg))

    ln0 = strlen((char *)ht0);
    memcpy(htx, ht0, ln0);
    rdfm (htx+ln0);
    memcpy(htx+38, htx+ln0, 4);

    lnx = strlen((char *) (htx+20)) ;
    utos(htx+12, lnx, 10, -1);
    utos(htx+ln0-8, lnx+20-ln0, 10, -1);
    htx[18] = 0;

    sendcmd(opn, ok);
    sendcmd(htx, prm);
    sendcmd(htx+20, rcv);
    sendcmd(cls, ok);

    wrs(pof);
    return 0;
}
```

```

}

unsigned char sendcmd(unsigned char *cmd, unsigned char *ir)
{
    qsec(4);
    wrs(cmd);
    return rdr(ir);
}

void qsec(unsigned char cnt)
{
    for (unsigned char i = cnt; i; i-- ) _delay_ms(250);
}

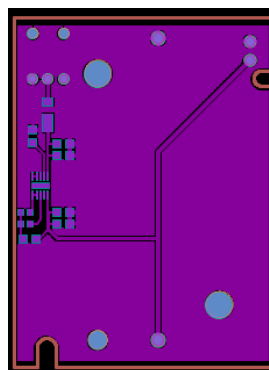
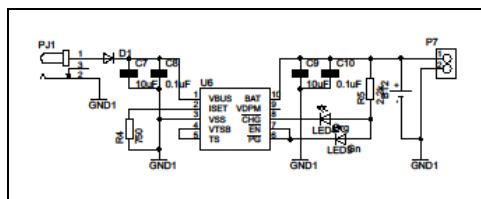
void wrs(const unsigned char *str)
{
    while (*str != '\0') txc(*str++);
}

unsigned char txc (unsigned char c)
{
    while ( !(UCSR0A & (1 << UDRE0)) );
    UDR0 = c;
    return c;
}

```

การพัฒนาแหล่งจ่ายไฟ

เนื่องจากการทำงานของ 3 จี โมดูล นั้นจะใช้กระแสไฟสูงเป็นช่วงเวลาที่สั้นๆ ในขณะที่เริ่มส่งข้อมูล ดังนั้นจึงต้องทำการออกแบบและสร้างวงจรแหล่งจ่ายไฟขึ้นใหม่ ดังแสดงในภาพที่ 3 ให้สามารถรองรับการทำงานดังกล่าว กระแสไฟจะได้มาจากแบตเตอรี่ลิเทียมไอออน และแผงเซลล์แสงอาทิตย์ขนาด 1 watts ซึ่งจะทำให้การชาร์ตแบตเตอรี่ ผ่านวงจรควบคุม ระบบดังกล่าวจะทำให้สถานีตรวจวัดอากาศแบบไร้สายไม่ต้องอาศัยไฟฟ้าจากแหล่งอื่นและติดตั้งได้สะดวก

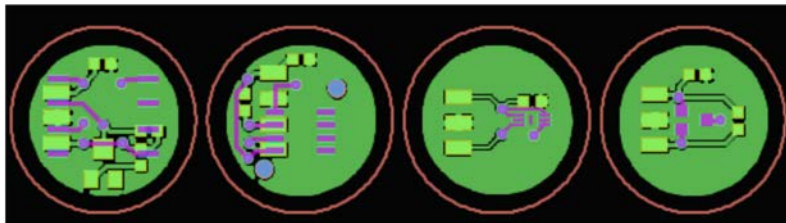
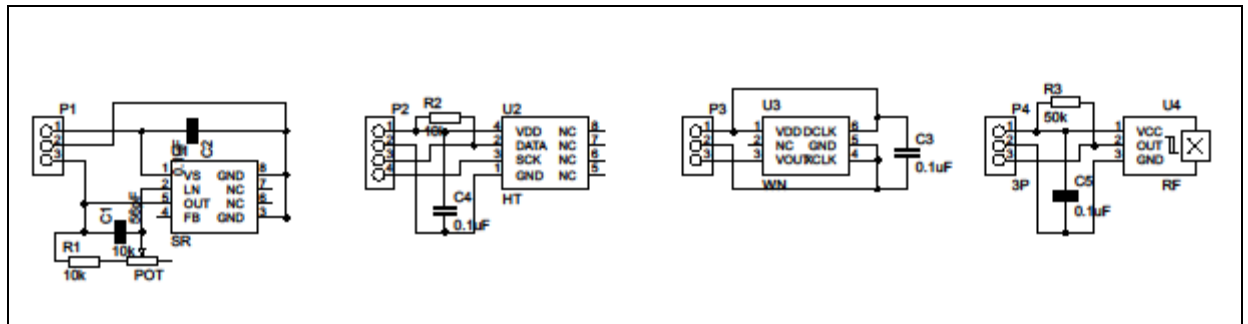


ภาพที่ 3 แผงวงจรและลายวงจร ของแหล่งจ่ายไฟฟ้า ภาพขนาดใกล้เคียงของจริง

การดำเนินงานในระยะที่ 2

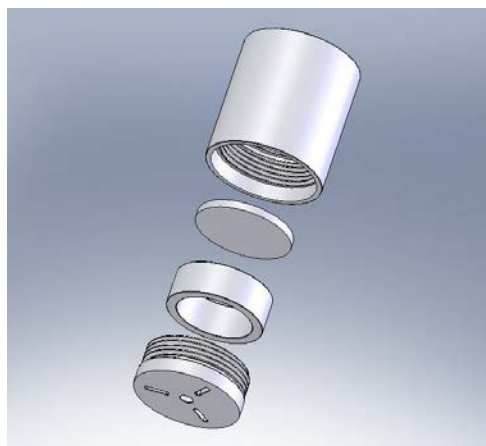
การสร้างเซ็นเซอร์ตรวจสอบสภาพอากาศ

ได้ออกแบบและสร้างเซ็นเซอร์ตรวจสอบสภาพอากาศ คือ แสงอาทิตย์ อุณหภูมิ ความชื้นอากาศ ความเร็วลม และปริมาณฝน ซึ่งประกอบด้วยส่วนที่เป็นวงจรอิเล็กทรอนิกส์ (ภาพที่ 4) และส่วนที่เป็นกลไกและอุปกรณ์บรรจุ (ภาพที่ 5 ก-) จำนวน 20 ชุด

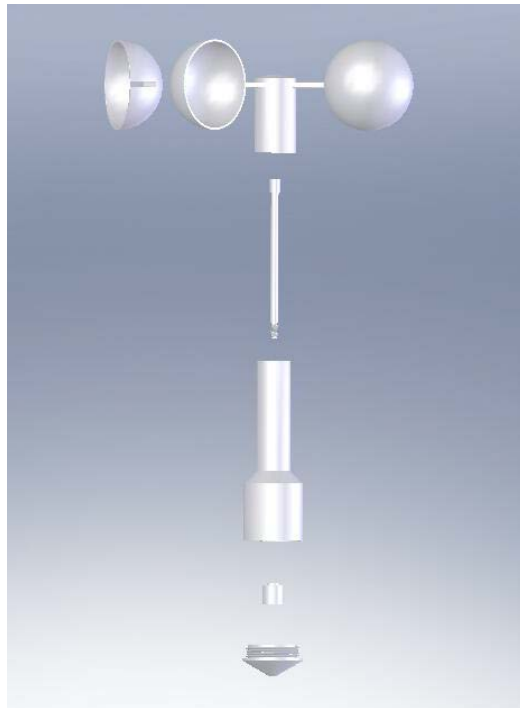


ภาพที่ 4 แผ่นผังวงจรอิเล็กทรอนิกส์ และแผ่นพิมพ์ของเซ็นเซอร์ วัดพลังแสงอาทิตย์ อุณหภูมิและความชื้นอากาศความเร็วลม และปริมาณฝน

เพื่อลดต้นทุนการผลิตจึงได้ออกแบบอุปกรณ์บรรจุเซ็นเซอร์วัดพลังแสงอาทิตย์ อุณหภูมิและความชื้นอากาศ และปริมาณฝน ให้มีขนาดเดียวกัน แต่ต่างกันที่รูเปิดด้านบนตามชนิดของเซ็นเซอร์เท่านั้น ดังแสดงในภาพที่ 5ก สำหรับเซ็นเซอร์วัดความเร็วลมมีชิ้นส่วนที่เป็นกลไกเพิ่มเติม ดังแสดงในภาพที่ 5ข

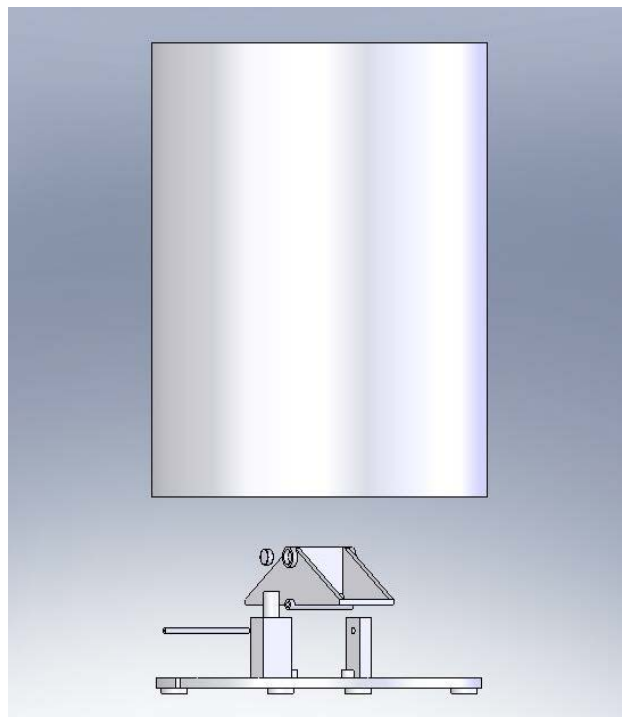


ภาพที่ 5ก แบบของกล่องที่ใช้บรรจุวงจรอิเล็กทรอนิกส์ของเซ็นเซอร์ วัดพลังแสงอาทิตย์ อุณหภูมิและความชื้นอากาศ และปริมาณฝน



ภาพที่ 5ข แบบของเซ็นเซอร์วัดความเร็วลม

สำหรับเซ็นเซอร์วัดปริมาณน้ำฝน นอกจากกล่องเพื่อใช้บรรจุแผงวงจรอิเล็กทรอนิกส์แล้วยังจำเป็นต้องมีส่วนที่เป็นกลไกคือกระบอกรับน้ำฝนและกระตือรือร้นวัด (tipping bucket) ดังแสดงในภาพที่ 5ค



ภาพที่ 5ค กระบอกรับน้ำฝนและกระตือรือร้นวัด (tipping bucket) ที่ต้องใช้ร่วมกับวงจรอิเล็กทรอนิกส์ของ sensor

การเทียบค่าของเซ็นเซอร์ (Sensor calibration)

เซ็นเซอร์ที่จะต้องทำการเทียบค่ากับอุปกรณ์ของ Campbell Scientific, Inc คือ เซ็นเซอร์วัดพลังแสงอาทิตย์ เนื่องจากสัญญาณที่ได้จากเซ็นเซอร์เป็นสัญญาณแบบอนาล็อก และความคลาดเคลื่อนยินยอมของเครื่องจักรที่ใช้ในการสร้างอาจมากพอที่จะก่อให้เกิดความแตกต่างของระดับสัญญาณได้

การเทียบค่าพลังแสงอาทิตย์ได้ใช้ Li-Cor LI200X pyranometer เป็นอุปกรณ์อ้างอิง โดยต่อเซ็นเซอร์ทั้ง 2 เข้ากับเครื่องบันทึกข้อมูล Campbell CR10X เพื่อไม่ให้มีความแตกต่างอื่นๆระหว่างเซ็นเซอร์ทั้งสอง ดังแสดงในภาพที่ 6 จากนั้นปรับความต้านทานบนวงจรอิเล็กทรอนิกส์ของเซ็นเซอร์ที่สร้างขึ้น จนกระทั่งความแตกต่างของค่าที่อ่านได้แบบเวลาจริงระหว่างเซ็นเซอร์ทั้งสองน้อยกว่า $\pm 5\%$



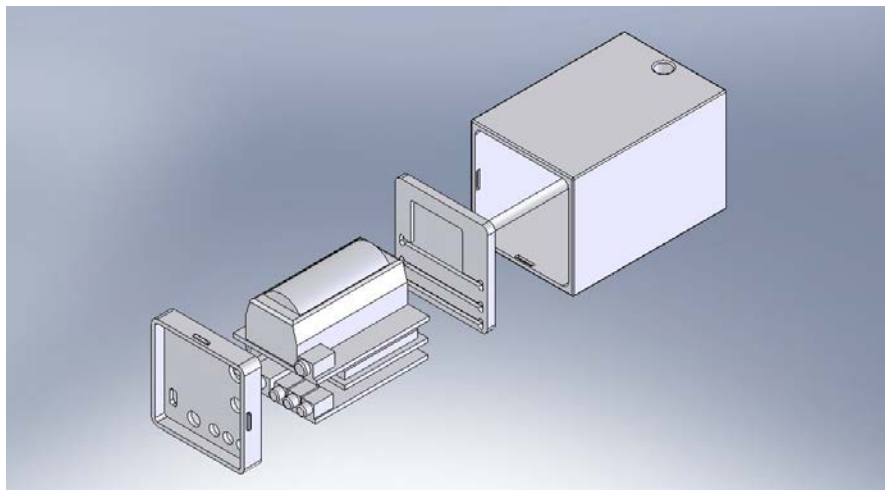
ภาพที่ 8 เทียบค่าพลังแสงอาทิตย์โดยใช้ Li-Cor LI200X pyranometer เป็นอุปกรณ์อ้างอิง โดยต่อเซ็นเซอร์ทั้ง 2 เข้ากับเครื่องบันทึกข้อมูล Campbell CR10X และอ่านค่าการวัดจากเครื่อง PC แบบเวลาจริง

การเทียบค่าปริมาณน้ำฝนทำได้โดยปรับความสูงต่ำของสกรูที่ด้านล่างของกระเบื้อง จนกระทั่งทำให้กระเบื้องกระดกเมื่อได้รับน้ำเป็นปริมาตรเท่ากับ 3.0 มล. หรือเทียบเท่ากับปริมาณฝน 0.23 มม.

ส่วนเซ็นเซอร์วัดอุณหภูมิและความชื้นนั้น วงจรรวมที่ใช้ได้รับการเทียบค่ามาแล้วจากบริษัทผู้ผลิต และได้รับการตรวจสอบอย่างละเอียดแล้วใน โครงการการพัฒนาเครื่องวัดสภาพภูมิอากาศอัตโนมัติพื้นฐาน เพื่อจัดเก็บข้อมูลภูมิอากาศแบบบูรณาการสำหรับสนับสนุนการตัดสินใจทางการเกษตรและการจัดการทรัพยากร สัญญาเลขที่ RDG53O0009 (เกริก 2555) เช่นเดียวกับเซ็นเซอร์วัดความเร็วลมที่มีมิติเท่ากับของเดิมที่ได้พัฒนาขึ้น (เกริก 2555) และได้ทำการเทียบค่ากับระบบของ Campbell ไว้แล้ว

การประกอบสถานีตรวจวัดสภาพอากาศ

เมื่อสร้างวงจรอิเล็กทรอนิกส์ของ 3 จี โมดูล เครื่องบันทึกข้อมูลและแหล่งจ่ายไฟเสร็จแล้วได้ออกแบบสร้างกล่องบรรจุวงจรเหล่านี้ไว้ด้วยกันดังแสดงในภาพที่ 9



ภาพที่ 9 แบบของกล่องบรรจุวงจร 3 จี โมดูล G module เครื่องบันทึกข้อมูล และแหล่งจ่ายไฟ

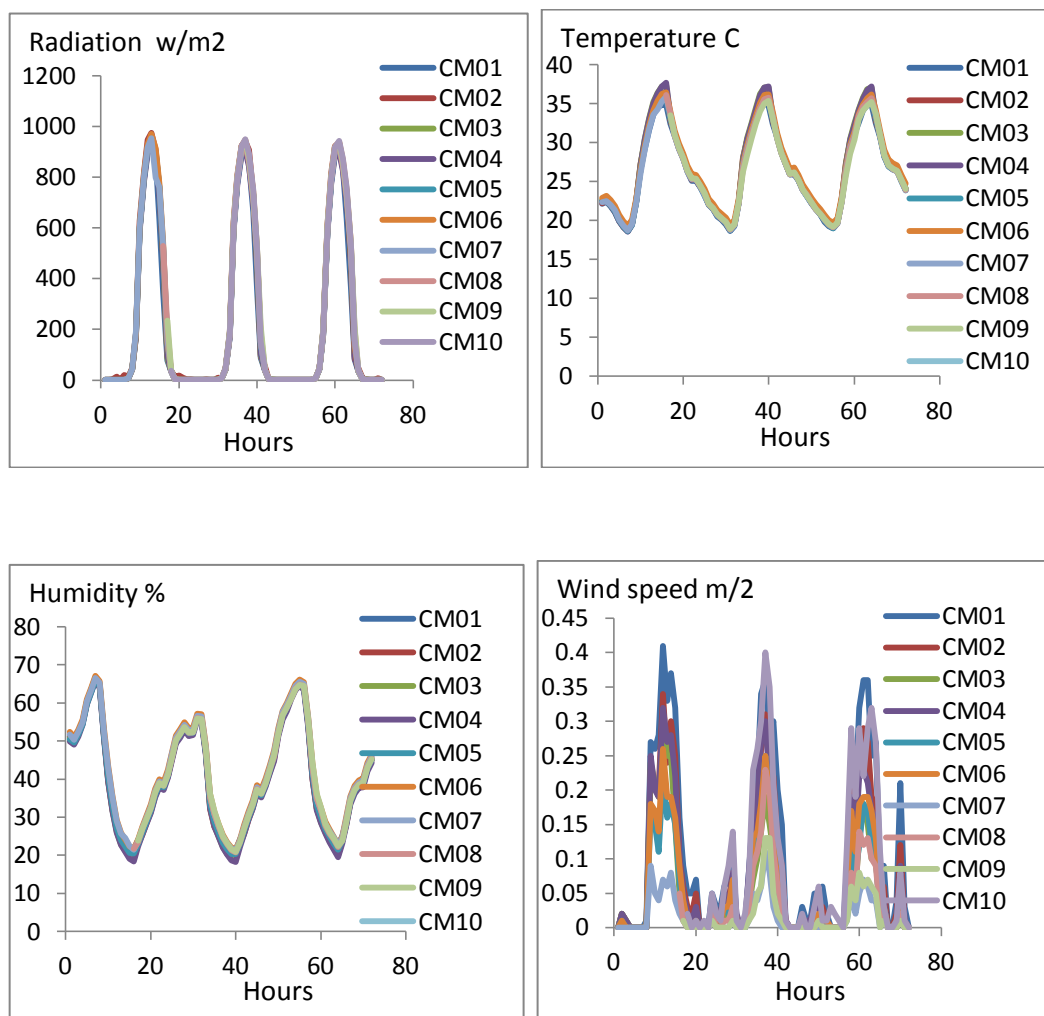
จากนั้นนำเซ็นเซอร์ 3 จี โมดูล เครื่องบันทึกข้อมูลและแหล่งจ่ายไฟมาประกอบเข้าด้วยกันเป็นสถานีตรวจวัดสภาพอากาศ (ภาพที่ 10) จำนวนทั้งหมด 20 ชุด



ภาพที่ 10 สถานีตรวจวัดสภาพอากาศ ที่ได้พัฒนาขึ้น

การตรวจสอบการทำงานและความแม่นยำของสถานีตรวจวัดสภาพอากาศเบื้องต้น

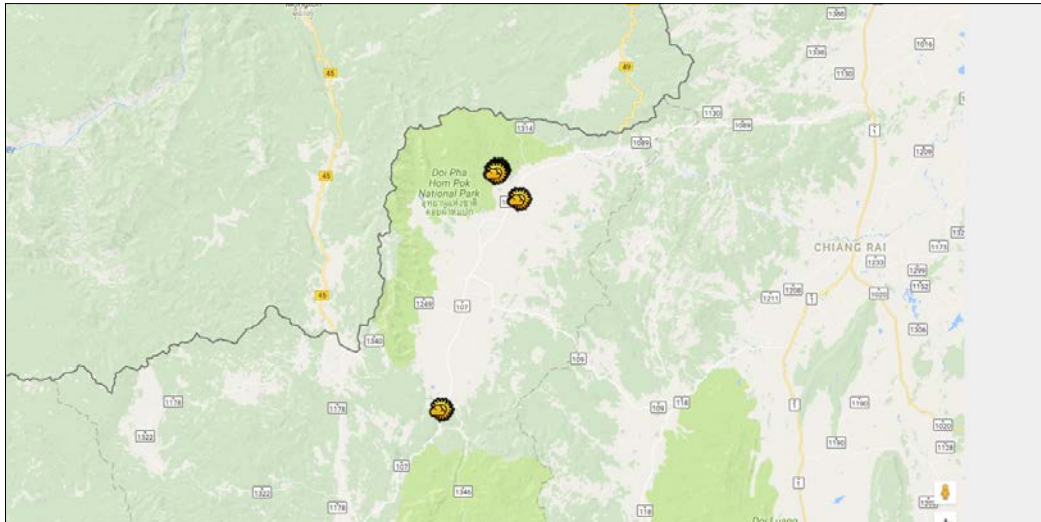
ได้ทดสอบเปรียบเทียบการทำงานและค่าที่ได้จากการวัดของสถานีตรวจวัดสภาพอากาศเบื้องต้นที่ คณะเกษตรศาสตร์ มหาวิทยาลัยขอนแก่นนั้น โดยใช้สถานีตรวจวัดสภาพอากาศจำนวน 10 ชุด ผลการ (ภาพที่ 12 ก ข ค และ ง) ซึ่งค่าการวัดของพลังงานแสงอาทิตย์ อุณหภูมิ ความชื้นสัมพัทธ์อากาศมีค่าใกล้เคียงกันมาก ความแตกต่างที่เกิดขึ้นเล็กน้อยนั้นเกิดจากความไม่สม่ำเสมอของสภาพแวดล้อมในบริเวณ ที่ทำการทดสอบ อย่างไรก็ตามค่าความเร็วลมจะแตกต่างกันอย่างเห็นได้ชัดโดยเฉพาะเครื่องหมายเลข cm08 และ cm09 ความแตกต่างดังกล่าวอาจเกิดจากความแตกต่างของความถี่ของตลับลูกปืนในช่วง เริ่มแรกของการใช้งาน ซึ่งจะแสดงออกชัดเจนในสภาพลมสงบ (ความเร็วลมต่ำกว่า 0.5 m/s)



ภาพที่ 12 ก ข ค และ ง เปรียบเทียบค่าการวัดพลังงานแสงอาทิตย์ อุณหภูมิ ความชื้นสัมพัทธ์อากาศ และความเร็วลมของเครื่องวัดสภาพอากาศหมายเลข CM01 ถึง CM10 ในระหว่างการทดสอบเบื้องต้นที่คณะเกษตรศาสตร์ มหาวิทยาลัยขอนแก่น

การติดตั้งและทดสอบภาคสนาม

ได้นำสถานีตรวจวัดสภาพอากาศจำนวน 10 ชุด ไปติดตั้งทำการทดสอบภาคสนามในสวนส้ม อำเภอฝาง และอำเภอไชยปราการ อำเภอแม่อาย จังหวัดเชียงใหม่ จำนวน 9 ชุดตามตำแหน่งที่ได้แสดงไว้บนแผนที่ในภาพที่ 11 และอีก 1 ชุด ที่ศูนย์วิจัยแม่เหียะ คณะเกษตรศาสตร์ มหาวิทยาลัย เชียงใหม่ ตามที่ สกว ได้มอบหมาย



ภาพที่ 11 ตำแหน่งที่ติดตั้ง weather station 9 เครื่องในสวนส้ม 3 แห่ง ในอำเภอฝาง อำเภอแม่อายและอำเภอไชยปราการ จังหวัดเชียงใหม่

การส่งข้อมูลของเครื่องวัดสภาพอากาศผ่านเครือข่าย 3 จี-อินเทอร์เน็ต ไปยังเว็บเซิร์ฟเวอร์

จากวันที่ทำการติดตั้งที่ อ.ฝาง อ. ไชยปราการ และ อ. แม่อาย และ อ. เมือง จ. เชียงใหม่ ในที่ 8 มีนาคม ถึงวันที่ 25 มีนาคม สถานีวัดสภาพอากาศสามารถส่งข้อมูลสภาพอากาศมายังเว็บเซิร์ฟเวอร์ ของศูนย์วิจัยระบบทรัพยากรเกษตร คณะเกษตรศาสตร์ มหาวิทยาลัยเชียงใหม่ ([www.mcc.cmu.ac.th/projects/iWeather Station](http://www.mcc.cmu.ac.th/projects/iWeather%20Station)) ได้อย่างถูกต้องและครบถ้วน อย่างไรก็ตามมีจำนวนข้อมูลประมาณ 4% ที่มีการส่งล่าช้ากว่าเวลาที่กำหนด ทั้งนี้เนื่องจากสถานีตรวจวัดสภาพอากาศไม่สามารถค้นหาและเชื่อมต่อกับเว็บเซิร์ฟเวอร์ได้ตามระยะเวลาที่กำหนดไว้ ดังนั้นเครื่องจึงสำรองข้อมูลนั้นไว้และส่งมาพร้อมกับข้อมูลชุดใหม่ในครั้งต่อไป ดังนั้นจึงได้ทดลองขยายระยะเวลาของการรอเพื่อให้เครื่องมีเวลาในการค้นหาและเชื่อมต่อกับเว็บเซิร์ฟเวอร์ได้นานขึ้น โดยยอมให้กระบวนการบันทึกข้อมูลซึ่งเกิดขึ้นทุกๆนาที่ สามารถขัดขวาง (interrupt) กระบวนการส่งข้อมูลได้ พร้อมกับได้เปลี่ยนคำสั่งให้ส่งข้อมูลสภาพอากาศรายชั่วโมงจากวันละครั้งมาเป็นทุกๆชั่วโมงเพื่อให้ข้อมูลบนเว็บเซิร์ฟเวอร์ใกล้เคียงเวลาจริงมากขึ้น

การดำเนินงานในระยะที่ 3

การทดสอบในช่วงฤดูฝนต่อเนื่องจากระยะที่ 2

ได้ติดตั้งสถานีตรวจวัดสภาพอากาศเพิ่มเติมอีกในสามพื้นที่คือ ที่ ต.บ้านใหม่ อ.เมือง จ.พะเยา ส่วนเคหะการเกษตร อ.ลาดหลุมแก้ว จ.ปทุมธานี และ ในไร่ร่องน อ.มวกเหล็ก จ.สระบุรี นอกจากนี้ปัญหาที่พบและได้แก้ไขแล้วดังรายงานในรายละเอียดในหัวข้อ **ปัญหาที่พบและการแก้ไข** สถานีวัดสภาพอากาศเครื่องทั้งหมดทำงานเป็นปกติตามที่กำหนด ไม่มีความเสียหายใดๆเกิดขึ้นไม่ว่าจะเป็นส่วนของวงจรจริง อิเล็กทรอนิกส์หรือส่วนที่เป็นกลไกหรือส่วนของชุดคำสั่งควบคุมการทำงานตลอดช่วงระยะเวลาที่ได้ทำการทดสอบ (วิธีการวิเคราะห์ปัญหาและแก้ไขได้รายงานในภาคผนวกที่ 2)

ปัญหาที่พบและการแก้ไข

1. ในการออกแบบได้ใช้ปลั๊กสัญญาณเสียงขนาด 2.5 มม. สำหรับการเชื่อมต่อสายสัญญาณระหว่างเครื่องบันทึกข้อมูลและเซ็นเซอร์ เพื่อป้องกันไม่ให้เกิดความผิดพลาด (สลับสาย) ในการต่อสายเมื่อทำการติดตั้งสถานี ซึ่งความผิดพลาดของการต่อสายดังกล่าวสามารถทำให้เกิดความเสียหายกับอุปกรณ์ต่างๆในวงจรอิเล็กทรอนิกส์ได้ แต่เมื่อผ่านการใช้งานในสภาพแปลงเป็นระยะเวลานานขึ้นพบว่าปลั๊กตัวเมียเกิดการคายตัว (plastic deform: creep/relaxation) ทำให้การรับส่งสัญญาณไม่สมบูรณ์และค่าที่เครื่องบันทึกข้อมูลได้รับจากเซ็นเซอร์คลาดเคลื่อน ดังนั้นจึงได้เปลี่ยนจากการใช้ปลั๊กสัญญาณเสียงมาเป็นแบบ male และ female header แทน และหลังจากนั้นไม่พบปัญหาดังกล่าวเกิดขึ้นอีก

2. การออกแบบกล่องบรรจุวงจรบันทึกข้อมูล 3จี โมดูล และแหล่งจ่ายไฟนั้น ได้พยายามให้สามารถติดตั้งภายนอกอาคารได้โดยตรงเพื่อให้สะดวกและรวดเร็วในการติดตั้งและบำรุงรักษาในอนาคต ด้านบนและด้านข้างของกล่องจะทำด้วยอลูมิเนียมไม่มีรอยต่อเพื่อช่วยลดความร้อนจากแสงอาทิตย์และเพื่อป้องกันการซึมเข้าของน้ำ ที่ด้านล่างจะมีขอบกันน้ำไม่ให้ไหลเข้าสู่ฝาปิดด้านล่าง ซึ่งมีช่องสำหรับเชื่อมต่อกับเซ็นเซอร์ แผงแผงเซลล์แสงอาทิตย์ และเสาอากาศ 3 จี ดังนั้นจึงสะดวกในการประกอบ/ถอดชิ้นส่วนในระหว่างการเคลื่อนย้ายและติดตั้งโดยใช้เวลาไม่เกิน 20 นาทีเท่านั้น แต่เมื่อได้ทดสอบการใช้งานในสภาพแปลงพบว่าในสภาพที่มีพายุฝนลมแรง การเคลื่อนไหวแบบปั่นป่วนของลมจะทำให้กล่องฝนสามารถปลิวเข้าสู่ฟาด้านล่างและก่อให้เกิดความคลาดเคลื่อนของสัญญาณในระหว่างนั้น ดังนั้นจึงต้องนำเครื่องบันทึกข้อมูลไปบรรจุไว้ในกล่องกันน้ำระดับ IP47 เช่นเดียวกับเครื่องบันทึกข้อมูลสภาพอากาศอื่นๆ เช่นเครื่องบันทึกข้อมูลสภาพอากาศของ Campbell Scientific Inc เป็นต้น หลังจากการแก้ไขก็ไม่พบปัญหาดังกล่าวเกิดขึ้นอีกแม้ในช่วงที่มีพายุพายุดีเปรสชัน

ข้อจำกัด

จากการทดสอบภาคสนามพบว่าหากติดตั้งในบริเวณที่ความแรงของสัญญาณ 3จี (2.1 GHz) มีค่าต่ำกว่า -95 dBm (ซึ่งถือว่าเกือบใช้ไม่ได้) การส่งข้อมูลจะไม่เสถียรและจะขึ้นกับอยู่กับสภาพอากาศ หากสภาพอากาศมีความชื้นต่ำไม่มีฝนการส่งข้อมูลจะเป็นปกติ แต่ในสภาพอากาศที่มีความชื้นสูงหรือมีฝนระดับความแรงของสัญญาณจะลดลง เนื่องจากโมเลกุลของน้ำจะดูดซับพลังงานของคลื่นแม่เหล็กไฟฟ้า

ในช่วงไมโครเวฟได้ดี (Chaplin 2017) และส่งผลทำให้การรับส่งสัญญาณไม่เสถียร หากไม่สามารถหลีกเลี่ยงการติดตั้งสถานีตรวจวัดสภาพอากาศในพื้นที่ที่มีความแรงของสัญญาณต่ำกว่า -90 dBm ได้ จำเป็นที่จะต้องใช้อากาศแบบทิศทางเดียว (uni-direction) ช่วยเพิ่มระดับการรับส่งสัญญาณ แต่โครงการนี้ไม่ได้ทดสอบการใช้เสาอากาศดังกล่าว

สรุปผลการดำเนินการ

ได้ออกแบบและสร้างสถานีตรวจวัดอากาศแบบไร้สายให้ส่งข้อมูลสภาพอากาศ ณ เวลาใกล้เคียงปัจจุบันจากภาคสนาม ผ่านเครือข่าย 3 จี และ อินเทอร์เน็ตไปเก็บไว้ยังเว็บเซิร์ฟเวอร์ ขึ้น 20 ชุด สถานีตรวจวัดอากาศดังกล่าวประกอบด้วยเซ็นเซอร์ วัดพลังแสงอาทิตย์ อุณหภูมิ ความชื้นอากาศ ความเร็วลม และปริมาณฝน เครื่องบันทึกข้อมูล 3 จี โมดูล และชุดให้พลังงานไฟฟ้า จากนั้นได้ติดตั้งทดสอบสถานีดังกล่าวในสวนส้มในอำเภอไชยปราการ ฝาง และแม่อาย จังหวัดเชียงใหม่ จำนวน 9 เครื่อง ที่ศูนย์วิจัยแม่เหียะ คณะเกษตรศาสตร์ มหาวิทยาลัย เชียงใหม่ 1 เครื่องที่ ต.บ้านใหม่ อ.เมือง จ.พะเยา 1 เครื่องที่ สวนเคหะการเกษตร อ.ลาดหลุมแก้ว จ.ประทุมธานี 1 เครื่อง และ อีก 1 เครื่องในไร่องุ่น อ.มวกเหล็ก จ.สระบุรี

สถานีตรวจวัดอากาศอัตโนมัติสามารถส่งข้อมูลสภาพอากาศไปยังเว็บเซิร์ฟเวอร์ ของศูนย์วิจัยระบบทรัพยากรเกษตร มหาวิทยาลัยเชียงใหม่ โดยไม่มีการสูญเสียข้อมูล ความแม่นยำของการวัดพลังแสงอาทิตย์ อุณหภูมิ ความชื้นอากาศ มีความแม่นยำสูง แต่สำหรับความเร็วลมอาจมีความแตกต่างระหว่างเครื่องภายใต้สภาพลมสงบ (<0.45 เมตรต่อวินาที) ซึ่งคาดว่าเกิดจากความผิดของตลับลูกปืนที่ไม่สม่ำเสมอในช่วงเริ่มแรกของการใช้งาน

เมื่อทำการทดสอบเป็นระยะเวลานานขึ้นในช่วงฤดูฝนพบว่าในบางกรณีปลั๊กเชื่อมต่อกับเซ็นเซอร์เกิดการคายตัวทำให้ค่าที่เครื่องบันทึกข้อมูลได้รับจากเซ็นเซอร์คลาดเคลื่อน จึงได้เปลี่ยนปลั๊กที่ใช้เชื่อมต่อเป็นชนิดอื่นพร้อมกับนำกล่องพลาสติกกันน้ำมาตรฐาน IP47 มาบรรจุเครื่องบันทึกข้อมูลเพื่อป้องกันละอองฝนในขณะที่มีพายุฝนลมแรงอีกชั้นหนึ่ง หลังการแก้ไขไม่พบปัญหาดังกล่าวอีก นอกจากนี้พบว่าควรหลีกเลี่ยงการติดตั้งสถานีในพื้นที่ๆมีระดับความแรงของสัญญาณต่ำกว่า -90 dBm เพื่อให้การเชื่อมต่อและส่งข้อมูลของสถานีมีเสถียรภาพสูง

เอกสารอ้างอิง

- เกริก บัณฑิตเพ็ชร 2555. โครงการการพัฒนาเครื่องวัดสภาพภูมิอากาศอัตโนมัติพื้นฐานเพื่อจัดเก็บข้อมูลภูมิอากาศแบบบูรณาการสำหรับสนับสนุนการตัดสินใจทางการเกษตรและการจัดการทรัพยากร (Development of a basic automatic weather station for integrated data collection to support decision-making in agriculture and resource management). รายงานวิจัยฉบับสมบูรณ์สำนักงานกองทุนสนับสนุนการวิจัย (สกว.) สัญญาเลขที่ RDG53O0009
- เกริก บัณฑิตเพ็ชร ไกรสร อัญชลีวรพันธุ์ รวิภัทร์ เสือโต สมศักดิ์ พิชัยบัณฑิตกุล 2557. โครงการการประเมินความสอดคล้องเบื้องต้นเครื่องวัดสภาพอากาศอัตโนมัติเพื่อผลิตเชิงพาณิชย์ (Precompliance testing of automatic weather station for commercial production) รายงานวิจัยฉบับสมบูรณ์สำนักงานกองทุนสนับสนุนการวิจัย (สกว.) สัญญาเลขที่ RDG 5650128
- Baggio, A. 2015. Wireless sensor networks in precision agriculture. Delft University of Technology, The Netherlands. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.120.46&rep=rep1&type=pdf>.
- Camilli, A., C. E. Cugnasca, A. M. Saraiva, A. R. Hirakawa, P.L.P. Correa. 2007. From wireless sensors to field mapping: Anatomy of an application for precision agriculture. *Computers and Electronics in Agriculture*. 58, 25–36.
- Chaplin Martin. 2017 Water and microwave. http://www1.lsbu.ac.uk/water/microwave_water.html
- Chaudhary, D.D, S.P.Nayse, L.M.Waghmare. 2011. Application of wireless sensor network for greenhouse parameter control in precision agriculture. *International Journal of Wireless & Mobile Networks (IJWMN)*.3(1), 140-149.
- Garcia-Sanchez, A, F. Garcia-Sanchez, J. Garcia-Haro. 2011. Wireless sensor network deployment for integrating video-surveillance and data-monitoring in precision agriculture over distributed crops. *Computers and Electronics in Agriculture*. 75, 288–303.
- Kassim, M.R.M., I. Mat, and A.N. Harun. 2014. Wireless Sensor Network in Precision Agriculture Application. 1-5. In *International conference on Computer, Information and Telecommunication Systems (CITS)*, IEEE Publication, ISBN 978-1-4799-4384-5.
- Keshtgary, M. and A Deljoo. 2012. An efficient wireless sensor network for precision agriculture. *Canadian Journal on Multimedia and Wireless Networks*. 3(1), 1-5.
- Liqiang, Z., Y. Shouyi, L. Leibo, Z. Zhen, W. Shaojun. 2011. A Crop Monitoring System Based on Wireless Sensor Network. *Procedia Environmental Sciences*.11, 558–565.
- Rehman, A.,A. Z. Abbasi, N. Islam, Z. Shaikh. 2014. A review of wireless sensors and networks' applications in agriculture. *Computer Standards & Interfaces*. 36, 263–270.
- Sakthipriya, N. 2014. An Effective Method for Crop Monitoring Using Wireless Sensor Network. *Middle-East Journal of Scientific Research*. 20(9), 1127-1132.
- Shining, L., C. Jin, and L. Zhigang. 2011. Wireless Sensor Network for Precise Agriculture Monitoring, 307-310. In *International Conference on Intelligent Computation Technology and Automation (ICICTA)*. Vol. 1, IEEE Publication, ISBN 978-1-61284-289-9.

Wang, N., N. Zhang, and M. Wang. 2006. Wireless sensors in agriculture and food industry—Recent development and future perspective. *Computers and Electronics in Agriculture*.50, 1–14.

ภาคผนวกที่ 1 ชุดคำสั่งระบบปฏิบัติการของสถานีตรวจวัดสภาพ

อากาศ

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include "uart.h"

void qsec(unsigned char);
void uartini(void);
void spiini(void);
void adcini(void);
void shtini(void);
void tcnini(void);
void irqini(void);
void rtcini(void);
void login(void);
void upld(void);

int main(void)
{
    qsec(2);
    DDRB = 0x00;
    PORTB = 0xFF;
    DDRC = 0x00;
    PORTC = 0xF6;
    DDRD = 0x00;
    PORTD = 0xFF;

    uartini();
    adcini();
    shtini();
    tcnini();
    spiini();
    irqini();
    rtcini();
    login();
    sei();

    while(1)
    {
        PRR = (1<<PRTWI)|(1<<PRTIM2)|(1<<PRSPI)|(1<<PRADC);
        SMCR = 0x01;
        asm volatile ("sleep::");
        upld();
    }
}

//m1.c
#include <avr/io.h>
#include <avr/interrupt.h>
#include "uart.h"
#include "udfm.h"
#include "urtc.h"
#include "uadc.h"
#include "usht.h"
#include "ulog.h"
#include "wtcp.h"

extern unsigned int wpa, wba, rpa, rba;
static unsigned int data [8];
```

```

void ulog(void);
void upld(void);

ISR (INT0_vect)
{
    PRR = (1 << PRIM2);
    rtc0();
    ulog();
}

void ulog(void)
{
    data[0] = rtc(YM);
    data[1] = rtc(DH);
    data[2] += adc0();
    data[3] += sht(TEMPERATURE) >> 4;
    data[4] += sht(HUMIDITY) >> 2;
    data[5] += TCNT1;
    data[6] += TCNT0;
    TCNT1 = 0;
    TCNT0 = 0;
    data[7] = rtc(MS);

    if ( ((unsigned char) (data[7] >> 8)) == 0x59)
        wrfm((unsigned char *) data, 14);
}

void upld(void)
{
    static unsigned int  spa = 0, sba = 0;
    static unsigned char state = 0;
    unsigned char mm = (unsigned char) (data[7] >> 8);

    if ((mm==ALM) || (state == 1)) state = tcp(state);

    switch (state)
    {
        case 0: break;
        case 1: break;
        case 2:
            state = 0;
            rba = sba;
            rpa = spa;
            break;
        case 3:
            state = 0;
            sba = rba;
            spa = rpa;
    }
}

//m2.c
#include <avr/io.h>
#include <string.h>
#include "uart.h"
#include "udfm.h"
#include "ulog.h"
#include "wtcp.h"
#include "urtc.h"

static unsigned char net[] = "ATE0;+CREG?\r\n";

```

```

static unsigned char opn[] =
"AT+CGSOCKCONT=1,\"IP\", \"\"APN\" \";+NETOPEN=\"TCP\",80;+TCPCONNECT=\"\"HOST\" \",80\r\n";

static unsigned char cls[] = "AT+NETCLOSE\r\n";
static unsigned char pof[] = "AT+CPOF\r\n";

static unsigned char ok [] = "OK";
static unsigned char reg[] = "0,1";
static unsigned char prm[] = ">";
static unsigned char rcv[] = "OK";

static unsigned char ht0[] = "AT+TCPWRITE= \r\n PUT
/projects/iWeatherStation/api/RawData?id=STID&pw=PWD HTTP/1.1\r\nHost:
HOST\r\nContent-Length: \r\n\r\n";

unsigned char tcp(unsigned char op)
{
    unsigned char htx[1000] = {0};
    unsigned int ln0, lnx;
    unsigned char i = 2;

    uarton;
    switch (op)
    {
        case 0:
            PORTC = 0xFE;
            qsec(2);
            PORTC = 0xF6;
            qsec(40);
        case 1:
            qsec(4);
            rxclr();
            qsec(4);
            if (sendcmd(net, reg)) {if (op) break; else return 1;}
        case 2:
            ln0 = strlen((char *)ht0);
            memcpy(htx, ht0, ln0);
            rdfm (htx+ln0);
            lnx = strlen((char *) (htx+20));
            utos(htx+12, lnx, 10, -1);
            utos(htx+ln0-8, lnx+20-ln0, 10, -1);
            htx[18] = 0;

            if (sendcmd(opn, ok)) break;
            if (sendcmd(htx, prm)) break;
            if (sendcmd(htx+20, rcv)) break;
        case 3:
            i = 3;
            sendcmd(cls, ok);
    }
    qsec(4);
    wrs(pof);
    uartoff;
    return i;
}

unsigned char sendcmd(unsigned char *cmd, unsigned char *ir)
{
    qsec(4);
    wrs(cmd);
    return rdr(ir);
}

```

```

void qsec(unsigned char cnt)
{
    for (unsigned char i = cnt; i; i-- );
}

void login(void)
{
    unsigned char htx[] = "AT+TCPWRITE= \r\n PUT
/projects/iWeatherStation/api/RawData?id="STID"&pw="PWD" HTTP/1.1\r\nHost:
"HOST"\r\nContent-Length: 16\r\n\r\nYYYYMM DDHH MMSS\r\n";
    unsigned char n = 0;
    unsigned int lnx;

    uarton;
    qsec(40);
    rxclr();
    while ((sendcmd(net, reg)) && (n++ < 60)) qsec(40);
    lnx = strlen((char *) (htx+20));
    utos(htx+12, lnx, 10, -1);
    utos(htx+lnx+20-16, rtc(YM), 16, 4);
    utos(htx+lnx+20-11, rtc(DH), 16, 4);
    utos(htx+lnx+20-6, rtc(MS), 16, 4);
    htx[18] = 0;
    sendcmd(opn, ok);
    sendcmd(htx, prm);
    sendcmd(htx+20, rcv);
    sendcmd(cis, ok);
    qsec(4);
    wrs(pof);
    uartoff;
}

//m3.c
#include <avr/io.h>
#include <string.h>
#include "uart.h"
#include "urtc.h"
#include "wtcp.h"

void setclk(void)
{
    unsigned char s[64];
    unsigned char data[7];
    unsigned char n = 20, i;

    qsec(20);
    PORTC = 0xFE;
    qsec(2);
    PORTC = 0xF6;
    qsec(40);

    uarton;
    rxclr();
    while ((sendcmd((unsigned char *) "ATE0;+CREG?\r\n", (unsigned char
*) "0,1")) && (n--)) qsec(40);

    qsec(4);
    wrs((unsigned char
*) "AT+CHTPSERV=\"ADD\", \"time.is\", 80, 1;+CHTPUPDATE\r\n");
    qsec(40);

    wrs((unsigned char *) "AT+CCLK?\r\n");
    rds(s);
}

```

```

    qsec(4);
    uartoff;

    c = strchr((const char *)s, '');
    for (n = 0, c++; n < 7; n++, c++) {
        if (n == 2) data[n++] = 1;
        data[n] = (*c++-48) << 4;
        data[n] += *c++-48;
    }
    for (i = 0, n = 6; i < 7; i++, n--) wrrtc(0x08+i, data[n]);

    wrrtc(0x10, 0x80);
    for (i = 0x11; i < 0x17; i++) wrrtc(i, 0);
}

void setepctrl(void)
{
    wrrtc(0x00, 0x91);
    while (rdrtc(0x03) & 0x80);
    wrrtc(0x30, 0x42);
    do {nopcycle(10000);} while (rdrtc(0x03) & 0x80);
    wrrtc(0x00, 0x99);
}

void setctrl(void)
{
    wrrtc(0x01, 0x01);
    wrrtc(0x02, 0x00);
    wrrtc(0x03, 0x00);
}

void rtcini(void)
{
    TWIini;
    setctrl();
    setclk();
    setepctrl();
    wrrtc(0x02, 0x00);
}

//m4.c
#include <util/twi.h>
#include "urtc.h"

unsigned char twis()
{
    TWCR = (1<<TWINT)|(1<<TWSTA)|(1<<TWEN);
    while (!(TWCR & (1<<TWINT)));
    return (TWSR & 0xF8);
}

unsigned char twit(unsigned char uc_data)
{
    TWDR = uc_data;
    TWCR = (1<<TWINT)|(1<<TWEN);
    while (!(TWCR & (1<<TWINT)));
    return (TWSR & 0xF8);
}

unsigned char twir(unsigned char ack_bit)
{
    TWCR = (1<<TWINT)|(1<<TWEN)|(ack_bit<<TWEA);
    while (!(TWCR & (1<<TWINT)));
    return TWDR;
}

```

```

}

void twip()
{
    TWCR = (1<<TWINT)|(1<<TWEN)|(1<<TWSTO);
}

unsigned char rdrtc(unsigned char addr)
{
    unsigned char data;

    twis();
    twit(SLAW);
    twit(addr);
    twip();
    twis();
    twit(SLAR);
    data = twir(NACK);
    twip();
    return data;
}

Void wrrtc(unsigned char addr, unsigned char data)
{
    twis();
    twit(SLAW);
    twit(addr);
    twit(data);
    twip();
}

unsigned int rtc(unsigned char addr)
{
    union {unsigned int x; unsigned char c[2];} u;
    u.c[0] = rdrtc(addr++);
    u.c[1] = rdrtc(addr);
    return u.x;
}

void rtc0(void)
{
    wrrtc(0x02, 0x00);
}

//m5.c
#include <avr/io.h>
#include "uart.h"
#include "udfm.h"

unsigned int wpa = 0, wba = 0, rpa = 0, rba = 0;

void spiini(void)
{
    DDRB |= (1<<PORTB5)|(1<<PORTB3)|(1<<PORTB2);
    SPCR |= (1<<SPE)|(1<<MSTR)|(1<<CPOL)|(1<<CPHA);
}

unsigned char spi (unsigned char ch)
{
    SPDR = ch;
    while (!(SPSR & (1 << SPIF)));
    return SPDR;
}

```

```

}

void wrfm (unsigned char *data, unsigned int i)
{
    static unsigned char opcode = 0x82;

    DFon;
    spi(opcode);
    spi(wpa >> 6 );
    spi((wpa << 2) | (wba >> 8));
    spi(wba);

    while (i) {
        i--;
        spi(*data);
        *data++ = 0;
        if (++wba > 527) {
            opcode ^= 0x07;
            wpa++;
            wba = 0;
            break;
        }
    }
    DFOff;
    nopcycle (9600);
    if (i) wrfm(data, i);
}

void rdfrm (unsigned char * m)
{
    union {unsigned int x[7]; unsigned char c[14];} u;

    DFon;
    spi(0x03);
    spi(rpa >> 6 );
    spi((rpa << 2) | (rba >> 8));
    spi(rba);

    for (unsigned char h = 0; h < 24; h++)
    {
        if ((rpa >= wpa) && (rba >= wba)) break;

        for (unsigned char i = 0; i < 14; i++) u.c[i] = spi(0xFF);
        if ((rba += 14) > 527) {rpa++;      rba -= 528;}

        for (unsigned char i = 0; i < 7; i++) m = utos(m, u.x[i], 16, 4);
        if (u.c[2] == 0x23) {*m++ = 0x0D; *m++ = 0x0A;};
    }
    DFOff;
}

//m6.c
#include <avr/interrupt.h>

void irqini(void)
{
    EICRA = 0x02;
    EIMSK = (1<<INT0);
}

//m7.c
#include <avr/io.h>
#include "uart.h"

```

```

void uartini(void)
{
    UBRR0H = (unsigned char) (ubrr >> 8);
    UBRR0L = (unsigned char) ubrr;
    UCSR0A |= (1<<0 << U2X0 );
    UCSR0C = (1<< UCSZ01) | (1 << UCSZ00);
}

unsigned char txc(unsigned char c)
{
    while ( !(UCSR0A & (1 << UDRE0)) );
    UDR0 = c;
    return c;
}

unsigned char rxc(void)
{
    unsigned long int i = TIMEOUT;

    while (!(UCSR0A & (1<<RXC0))) if (--i == 0 ) return 0;
    return (UDR0);
}

unsigned char rxclr(void)
{
    unsigned char clr = UDR0;
    while (UCSR0A & (1<<RXC0)) clr = UDR0;
    return clr;
}

void wru(unsigned int u)
{
    if (u >= 10) wru(u / 10);
    txc((u % 10) + '0');
}

void wrun(unsigned int u, unsigned char n)
{
    if (n > 1) wrun(u/10, n-1);
    txc((u%10) + '0');
}

void wrxn(unsigned int u, unsigned char n)
{
    if (n > 1) wrxn(u >> 4, n-1);
    txc("0123456789ABCDEF"[(u & 0x000f)]);
}

void wrs(const unsigned char *str)
{
    while (*str != '\0') txc(*str++);
}

void rds(unsigned char *str)
{
    unsigned int i = 0;

    while ((*str++ = rxc()) && (++i < 64));
}

unsigned char rdr(const unsigned char *s0)
{

```

```

const unsigned char *s = s0;
unsigned char c;
unsigned int i = 1;

while (*s) {
    if (!(c = rxc()) || (++i > 512)) return i;
    if (*s++ != c) s = s0;
}
return 0;
}

void nopcycle (unsigned int i)
{
    while(i--) asm volatile ("nop::");
}

void sec (unsigned int s)
{
    while (s--)
        for (unsigned int i = 0; i < 1000; i++) nopcycle(530);
}

unsigned char * utos(unsigned char * s, unsigned int v, unsigned char base, signed
char fw)
{
    if ((fw > 1) || (v >= base)) s = utos(s, v/base, base, --fw);
    *s = "0123456789ABCDEF"[v % base];
    return ++s;
}

//m8.c
#include "usht.h"

void shtini(void)
{
    unsigned char i;
    SCK;
    DATA_OUT; asm volatile ("nop::");
    DATA1;
    SCK0;

    for (i = 9; i; i--)
    {
        SCK1;  asm volatile ("nop::");
        SCK0;  asm volatile ("nop::");
    }
    s_trans_start();
}

void s_trans_start(void)
{
    DATA_OUT;
    asm volatile ("nop::");
    DATA1;
    SCK0;
    asm volatile ("nop::");
    SCK1;
    asm volatile ("nop::");
    DATA0;
    asm volatile ("nop::");
    SCK0;
}

```

```

    asm volatile ("nop::");
    asm volatile ("nop::");
    asm volatile ("nop::");
    SCK1;
    asm volatile ("nop::");
    DATA1;
    asm volatile ("nop::");
    SCK0;
}

unsigned char s_write_byte(unsigned char value)
{
    unsigned char i, error=0;
    DATA_OUT;
    asm volatile ("nop::");

    for (i = 0x80; i; i >>= 1)
    {
        if (i & value)
            DATA1;
        else
            DATA0;
        SCK1;
        asm volatile ("nop::");
        asm volatile ("nop::");
        asm volatile ("nop::");
        SCK0;
    }

    DATA_IN;
    DATA1;
    asm volatile ("nop::");
    SCK1;
    asm volatile ("nop::");
    asm volatile ("nop::");
    asm volatile ("nop::");
    error = DATA_;
    SCK0;
    return error;
}

unsigned char s_read_byte(unsigned char ack)
{
    unsigned int i, val=0;
    DATA_IN;
    DATA1;
    asm volatile ("nop::");

    for (i = 0x80; i; i >>= 1)
    {
        SCK1;
        if (DATA_)
            val = (val|i);
        SCK0;
    }

    DATA_OUT;
    asm volatile ("nop::");

    if(ack)
        DATA0;
    else
        DATA1;
    SCK1;

```

```

    asm volatile ("nop::");
    asm volatile ("nop::");
    asm volatile ("nop::");
    SCK0;
    return val;
}

unsigned int sht(unsigned char cmd)
{
    unsigned char error;
    unsigned int i, msb, lsb;
    error=0;

    s_trans_start();
    error += s_write_byte(cmd);

    for (i = 10000; DATA_ && i; i--) _delay_loop_1(40);
    if (!i) error += 1;

    msb = s_read_byte(ACK);
    lsb = s_read_byte(ACK);
    return (msb<<8)|lsb;
}

//m10.c
#include "uadc.h"
void adcini(void)
{
    ACSR = (1<<ACD);
    DIDR0 = 0b00000001;
    ADCSRA = 0b00000101;
}

unsigned int adc0(void)
{
    ADCSRA |= 0b11000000;
    while ((ADCSRA & 0b00010000) == 0);
    ADCSRA &= 0b00010101;
    return ADC;
}

//m11.c
#include <avr/io.h>

void tcnini(void)
{
    TCCR1B = (1<<CS12)|(1<<CS11)|(0<<CS10);
    TCNT1 = 0;
    DDRD &= ~0x20;

    TCCR0B = (1<<CS12)|(1<<CS11)|(0<<CS10);
    TCNT0 = 0;
    DDRD &= ~0x10;
}

```

ภาคผนวกที่ 2 ขั้นตอนการวิเคราะห์และแก้ไขปัญหา

หากเครื่องวัดสภาพอากาศ ไม่สามารถส่งข้อมูลไปยังเซิร์ฟเวอร์ติดต่อกันมากกว่า 5 ครั้ง ปัญหาอาจเกิดจาก

1. แบตเตอรี่
2. แผงวงจรของ 3G โมดูล หรือ
3. แผงวงจรของไมโครคอนโทรลเลอร์

การตรวจสอบแบตเตอรี่

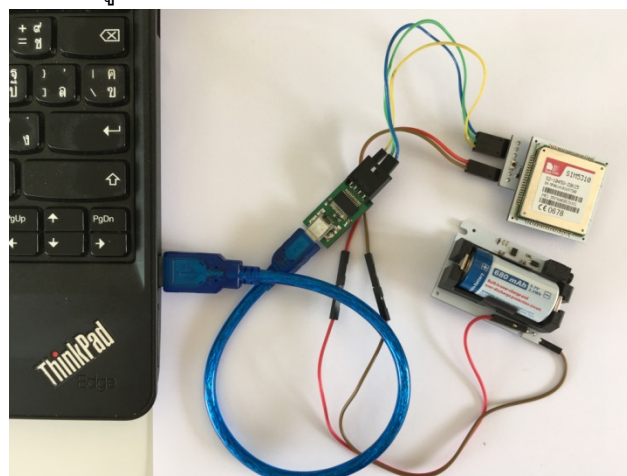
แบตเตอรี่ลิเทียมไอออนที่พร้อมใช้นั้นควรมีค่าแรงดันไฟฟ้าระหว่าง 3.7-4.0 V ให้ใช้มัลติมิเตอร์วัดแรงดันไฟฟ้าของแบตเตอรี่ หากค่าแรงดันไฟต่ำกว่านี้ให้ทำการชาร์จแบตเตอรี่โดยใช้เครื่องชาร์จแบตเตอรี่ที่มีคุณภาพ พร้อมทั้งตรวจสอบการทำงานของแผงโซลาร์และวงจรควบคุมแรงดันไฟฟ้าโดยวัดแรงดันไฟฟ้าที่ขั้วของรังแบตเตอรี่ภายใต้สภาพความเข้มแสงอาทิตย์ที่สูง แรงดันไฟฟ้าควรมีค่ามากกว่า 3.7 V และแผงวงจรจะต้องแสดงไฟสีเขียวและส้ม หากมีความบกพร่องของส่วนหนึ่งส่วนใดให้ทำการเปลี่ยนหรือซ่อมแซม

การตรวจสอบแผงวงจร 3G โมดูล

การตรวจสอบการทำงานของ 3G โมดูลโดยใช้ คำสั่ง AT นั้นเป็นการทดสอบการทำงานของทั้งระบบ เป็นการทดสอบเบื้องต้น และไม่สามารถระบุในรายละเอียดได้ว่าองค์ประกอบไหนของโมดูลเสีย

การตรวจสอบโดยใช้ คำสั่ง AT มีขั้นตอนดังนี้

1. ทำการเชื่อมต่อ 3G โมดูลเข้ากับ USB ของเครื่องพีซี PC โดยใช้อุปกรณ์แปลงสัญญาณ TTL-USB ดังแสดงในรูป



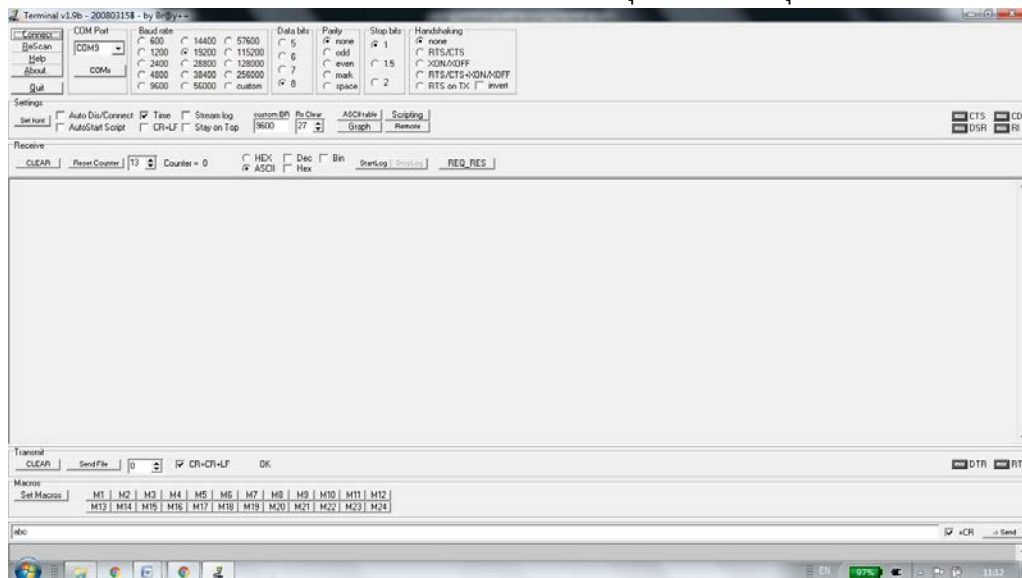
หากยังไม่เคยใช้อุปกรณ์ดังกล่าวมาก่อน ต้องทำการติดตั้งไดรฟ์เวอร์สำหรับอุปกรณ์ดังกล่าว โดยดาวน์โหลดไฟล์ **CDMv2.12.10WHQLCertified.exe** และ **AN_396 FTDI Drivers Installation Guide for Windows 10.pdf** จาก www.ftdichip.com/FTDrivers.htm จากนั้นให้คลิกขวาที่

CDMv2.12.10WHQLCertified.exe และ run as admin

เมื่อทำการติดตั้งไดรฟ์เวอร์เสร็จแล้วให้ต่ออุปกรณ์แปลงสัญญาณเข้ากับช่อง USB ของเครื่องพีซี จากนั้นให้เข้าไปที่ device manager | ports จะเห็น **USB Serial Port (COMx)** ซึ่ง x คือ

หมายเลขของ comport ที่อุปกรณ์แปลงสัญญาณใช้ (ให้ใช้ช่อง USB เดิมทุกครั้ง หากเปลี่ยนช่อง USB หมายเลขของ comport อาจเปลี่ยนได้)

2. ดาวน์โหลดโปรแกรม Bray terminal.exe จาก <https://sites.google.com/site/terminalbpp/> ดับเบิลคลิกโปรแกรม terminal.exe และกำหนดคุณสมบัติในการรับส่งข้อมูลของ bray terminal ดังนี้ 1. กำหนดค่า COM port ตามที่พบใน device manager, 2. Baud rate = 19200, 3. Data bit = 8, 4. Parity = None และ 5. Stop bits = 0 จากนั้นให้คลิกปุ่ม Connect ที่มุมบนด้านซ้าย



3. คลิกปุ่ม Set Macro ที่ด้านล่างซ้ายมือของหน้าต่าง terminal จากนั้นคลิก Load หากมีข้อความใดๆอยู่ในช่อง Transmit Macros ให้ลบข้อความเหล่านั้นออกให้หมด และเขียนคำสั่งแต่ละบรรทัดดังแสดงข้างล่างนี้ลงในช่อง M1, M2, M3, M4 และ M5 ของหน้าต่าง Macro Setting ดังที่แสดงในรูปและสั่งบันทึกในชื่อ Terminal_Modem.tmf

M1

ATE1;+CREG?#013#010

M2

AT+CHTTPSERV="ADD","time.is",80,1 #013#010

M3

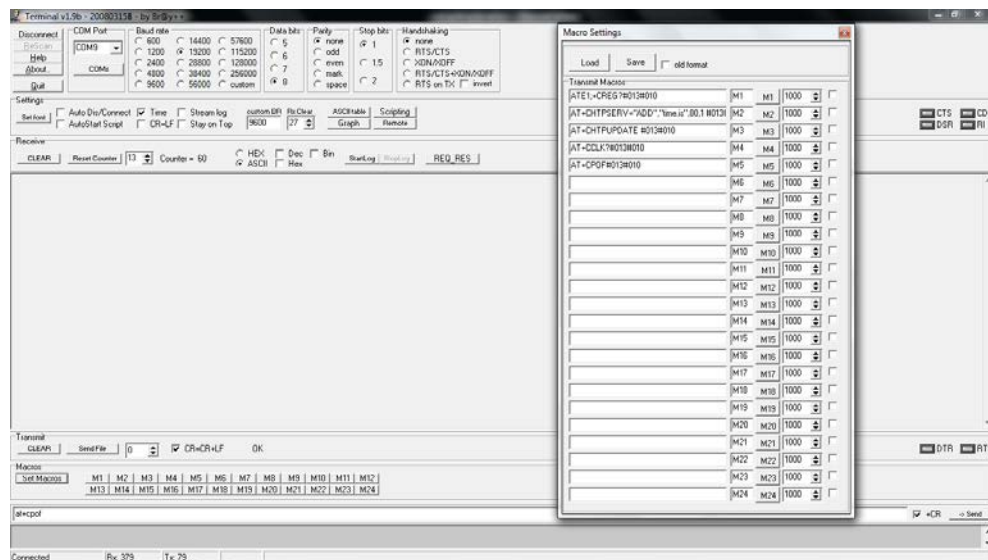
AT+CHTTPUPDATE #013#010

M4

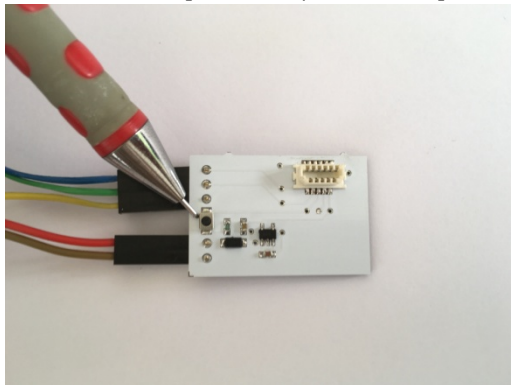
AT+CCLK?#013#010

M5

AT+CPOF#013#010



4. เปิดสวิตช์ 3G โมดูล โดยกดปุ่มดังแสดงในรูปเป็นเวลาสั้นๆ (1/2 วินาที)



5. หากไฟสีเขียวที่ด้านข้างของ 3G โมดูลจะสว่างขึ้น และหลังจากนั้น ประมาณ 10 วินาทีไฟก็กระพริบ แสดงว่า โมดูลพร้อมที่จะรับคำสั่ง แต่หากรอมากกว่า 2 นาที ก็ยังไม่มีไฟสีเขียวหรือไฟไม่กระพริบแสดงว่าองค์ประกอบส่วนใดส่วนหนึ่งของวงจรไม่ทำงานให้ทำการเปลี่ยนหรือซ่อมแซมหรือสัญญาณอาจจะอ่อนมากๆ
6. เมื่อ 3G โมดูลพร้อมทำงานให้คลิกปุ่ม M1 ที่ด้านล่างของ terminal เพื่อส่งคำสั่ง ATE0;+CREG? ไปยัง 3G โมดูล หากโมดูลทำงานเป็นปกติจะตอบกลับด้วย +CREG: 0,1 และ OK
7. เมื่อผ่านขั้นตอนที่ 6 ให้คลิกปุ่ม M2 เพื่อส่งคำสั่ง AT+CHTTPSERV="ADD","time.is",80,1 หากโมดูลทำงานเป็นปกติจะตอบกลับด้วย OK
8. หากผ่านขั้นตอนที่ 7 ให้คลิกปุ่ม M3 เพื่อส่งคำสั่ง AT+CHTTPUPDATE โมดูลที่ทำงานเป็นปกติจะตอบกลับด้วย OK และ+CHTTPUPDATE: 0
9. หากโมดูลทำงานปกติในขั้นตอนที่ 8 ให้คลิกปุ่ม M4 เพื่อส่งคำสั่ง AT+CCLK?#013#010 โมดูลที่ทำงานเป็นปกติจะตอบกลับด้วยเวลา เช่น +CCLK: "17/04/28,12:05:37+28" เป็นต้น
10. คลิกปุ่ม M5 เพื่อปิด 3G โมดูล

หาก 3G โมดูลแสดงการทำงานที่เป็นปกติทุกขั้นตอน แสดงว่าปัญหาอาจจะเกิดจากวงจรไมโครคอนโทรลเลอร์

(รายละเอียดของคำสั่ง AT สามารถดาวน์โหลดได้จาก
<http://simcomm2m.com/En/service/down.aspx>)

การตรวจสอบแพลงจอร์ของไมโครคอนโทรลเลอร์

การตรวจสอบการทำงานของวงจรมิโครคอนโทรลเลอร์โดยใช้คำสั่ง AT ก็เป็นการทดสอบการทำงานของทั้งระบบเช่นกัน และไม่สามารถระบุในรายละเอียดได้ว่าองค์ประกอบไหนของวงจรเสียหาย ขั้นตอนการดำเนินการก็เหมือนกันกับการตรวจสอบ 3G โมดูล เพียงแต่ในกรณีนี้ วงจรมิโครคอนโทรลเลอร์จะส่งคำสั่ง และ terminal จะทำหน้าที่ตอบกลับ

1. ให้เชื่อมต่อวงจรมิโครคอนโทรลเลอร์เข้ากับช่อง USB ของพีซีโดยใช้อุปกรณ์แปลงสัญญาณ TTL-USB (ในกรณีนี้ต้องใช้สายสัญญาณสีเขียวและสีน้ำเงิน) จากนั้นคลิกปุ่ม Set Macro ที่ด้านล่างซ้ายมือของหน้าต่าง terminal และคลิก Load หากมีข้อความใดๆ อยู่ในช่อง Transmit Macros ให้ลบข้อความเหล่านั้นออกให้หมด และเขียนคำสั่งแต่ละบรรทัดดังแสดงข้างล่างนี้ลงในช่อง M1, M2, M3 จนถึง M8 ของหน้าต่าง Macro Setting และบันทึกโดยใช้ชื่อ

Terminal_Controller.tmf

```
M1
+CREG: 0,1#013#010OK
M2
OK#013#010+CHTUPDATE: 0#013#010
M3
+CCLK: "17/03/17,11:32:06+28"#013#010OK#013#010
M4
+CREG: 0,1#013#010OK
M5
OK#013#010
M6
>
M7
204 No Content#013#010
M8
OK#013#010
```

2. ต่ออุปกรณ์แปลงสัญญาณ TTL-USB เข้ากับพีซี และคลิก connect บนหน้าต่างของ terminal
3. ต่อวงจรมิโครคอนโทรลเลอร์เข้ากับอุปกรณ์แปลงสัญญาณและต่อแบตเตอรี่เป็นขั้นตอนสุดท้าย
4. หลังจากเชื่อมต่อแบตเตอรี่ประมาณ 20 วินาที ไมโครคอนโทรลเลอร์จะส่งคำสั่ง ATE0;+CREG? มายัง terminal ให้คลิกปุ่ม M1 ทันที เพื่อให้ terminal ส่งคำตอบ +CREG: 0,1 และ OK ไปให้ ไมโครคอนโทรลเลอร์ หากรอเป็นเวลานาน (1 นาที) ก็ยังไม่เห็นคำสั่งดังกล่าวแสดงว่าวงจรมิโครคอนโทรลเลอร์อาจเสียหาย ซึ่งอาจเกิดจากส่วนที่เป็นซอฟต์แวร์หรือส่วนที่เป็นฮาร์ดแวร์ก็ได้

5. เมื่อได้รับคำตอบจากขั้นตอนที่ 4 ไมโครคอนโทรเลอร์ จะส่งคำสั่ง
AT+CHTTPSERV="ADD","time.is",80,1; +CHTTPUPDATE มายัง terminal ให้คลิกปุ่ม M2 ทันที เพื่อให้ terminal ส่งคำตอบ OK และ +CHTTPUPDATE: 0 ไปให้ไมโครคอนโทรเลอร์
6. หากผ่านขั้นตอนที่ 5 ไมโครคอนโทรเลอร์ จะส่งคำสั่ง AT+CCLK? มายัง terminal ให้คลิกปุ่ม M3 ทันที เพื่อให้ terminal ส่งเวลา +CCLK: "17/03/22,12:05:37+28" ไปให้ไมโครคอนโทรเลอร์
7. หลังจากที่ไม่โครคอนโทรเลอร์ได้รับเวลา ก็จะเริ่มทำการกำหนดการทำงานของส่วนต่างๆ บนวงจรเช่น นาฬิกา เป็นต้น จากนั้นจะเริ่มติดต่อกับเซิร์ฟเวอร์ เมื่อเห็นคำสั่ง ATE0;+CREG? อีกครั้ง ให้คลิกปุ่ม M4 เพื่อให้ terminal ส่งคำตอบ +CREG: 0,1 และ OK ไปให้ไมโครคอนโทรเลอร์
8. หากผ่านขั้นตอนที่ 7 ไมโครคอนโทรเลอร์ จะส่งคำสั่ง AT+CGSOCKCONT=1,"IP","internet"; +NETOPEN="TCP",80; +TCPCONNECT="www.xxx.yyy.ac.th",80 มายัง terminal ให้คลิกปุ่ม M5 ทันที เพื่อให้ terminal ส่งคำตอบ OK ไปให้ไมโครคอนโทรเลอร์
9. หากผ่านขั้นตอนที่ 8 ไมโครคอนโทรเลอร์ จะส่งคำสั่ง AT+TCPWRITE=203 มายัง terminal ให้คลิกปุ่ม M6 ทันที เพื่อให้ terminal ส่งเครื่องหมาย ">" ไปให้ไมโครคอนโทรเลอร์
10. จากนั้นไมโครคอนโทรเลอร์จะส่งข้อมูลไปให้เซิร์ฟเวอร์ และเมื่อเห็นการส่งข้อมูลยุติลงให้คลิก M7
11. จากนั้นไมโครคอนโทรเลอร์จะยุติการเชื่อมต่อโดยส่งคำสั่ง AT+NETCLOSE และต้องตอบด้วยการคลิก M8 เพื่อส่งคำตอบ OK

หากไมโครคอนโทรเลอร์ไม่สามารถทำงานได้ครบทุกขั้นตอนแสดงว่าวงจรไมโครคอนโทรเลอร์อาจเสียหาย ซึ่งอาจเกิดจากส่วนที่เป็นซอฟต์แวร์หรือส่วนที่เป็นฮาร์ดแวร์ก็ได้ ดังนั้นจำเป็นที่จะต้องทำการอัปเดตโปรแกรมควบคุมการทำงานของไมโครคอนโทรเลอร์เข้าไปใหม่ และตรวจสอบการทำงานทั้ง 11 ขั้นตอนอีกครั้ง เพื่อเป็นการตรวจสอบว่าปัญหาเกิดจากซอฟต์แวร์หรือไม่ ขั้นตอนของการการแปลงโปรแกรมควบคุมการทำงานของไมโครคอนโทรเลอร์จากภาษาซี ให้เป็น hex ไฟล์ โดยใช้ Atmel Studio และอัปเดตโปรแกรมเข้าสู่ไมโครคอนโทรเลอร์จะกล่าวในรายละเอียดในบทต่อไป

หากทำการอัปเดตโปรแกรมควบคุมการทำงานของไมโครคอนโทรเลอร์เข้าไปใหม่และดำเนินการตรวจสอบการทำงานทั้ง 11 ขั้นตอนอีกครั้งแล้ว แต่ยังไม่พบว่าการไมโครคอนโทรเลอร์ยังไม่ทำงานเป็นปกติ แสดงว่าฮาร์ดแวร์อาจเสียหาย ซึ่งขั้นตอนการตรวจสอบไอซีไมโครคอนโทรเลอร์ และไอซีนาฬิกาจะกล่าวในรายละเอียดในบทที่ 3

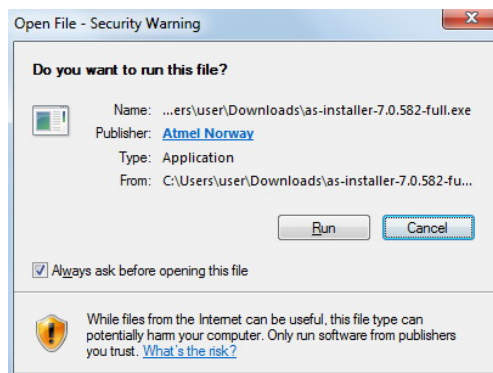
การแปลงโปรแกรมคอมพิวเตอร์การทำงานของไมโครคอนโทรลเลอร์จากภาษาซี ให้เป็น hex ไฟล์ โดยใช้ Atmel Studio

หากมี hex ไฟล์ของเครื่องตรวจสภาพอากาศเครื่องนั้นอยู่แล้วก็ไม่จำเป็นที่จะต้องแปลงโปรแกรม แต่สามารถ อัปโหลด hex ไฟล์ที่มีอยู่นั้นเข้าสู่ไมโครคอนโทรลเลอร์ได้เลย ขั้นตอนของการอัปโหลดจะอธิบายในบทถัดไป

การติดตั้ง Atmel Studio ในเครื่องพีซี

หากได้ทำการติดตั้ง Atmel Studio ไว้แล้วให้ข้ามหัวข้อนี้ไปหาหัวข้อต่อไป คือการแปลงโปรแกรมการทำงานของไมโครคอนโทรลเลอร์จากภาษาซี ให้เป็น hex ไฟล์

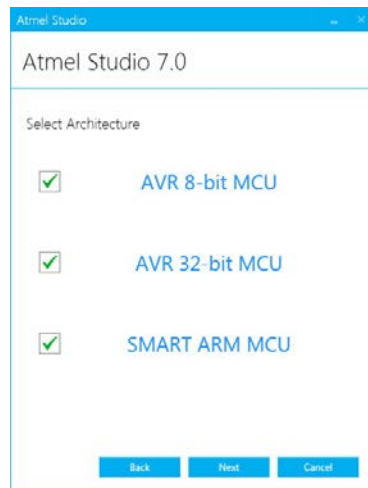
1. ดาวน์โหลด Atmel Studio **Atmel Studio 7.0 (build 1188) offline installer** จาก Atmel Website <http://www.atmel.com/tools/atmelstudio.aspx> ซึ่งเป็นการติดตั้งที่ไม่ต้องอาศัยการเชื่อมต่อกับ Internet
2. ดับเบิลคลิกไฟล์ที่ได้ดาวน์โหลดมาเพื่อเริ่มทำการติดตั้ง และคลิก **Run** ดังแสดงในภาพ



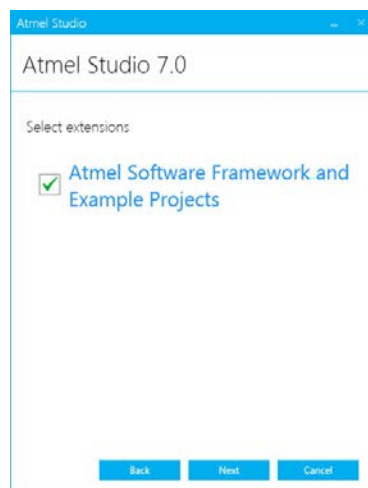
3. จากนั้นคลิกตกลงยอมรับเงื่อนไข และคลิก Next



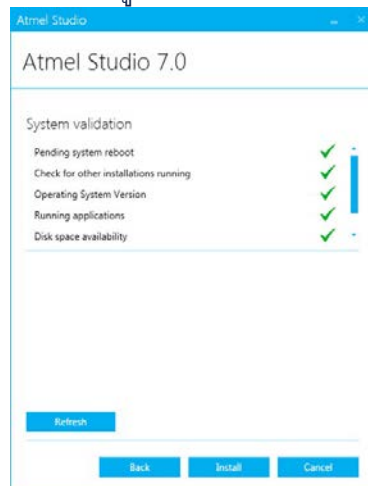
4. Weather station ใช้ AVR 8-bit MCU ฉะนั้นในช่อง select architecture เลือกแค่ AVR 8-bit MCU ก็เพียงพอ ไม่จำเป็นต้องเลือกทั้งหมด จากนั้นคลิก Next



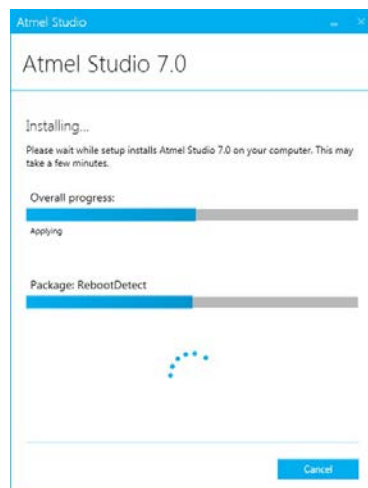
5. หากต้องการที่จะประหยัดพื้นที่บน hard disk ให้คลิกลบเครื่องหมายถูกจาก select extensions และ คลิก Next



เมื่อโปรแกรมได้ตรวจสอบว่า PC ที่ใช้มีคุณสมบัติเพียงพอที่สามารถรองรับการทำงานได้ก็จะแสดงผลดังรูป ให้คลิก Install



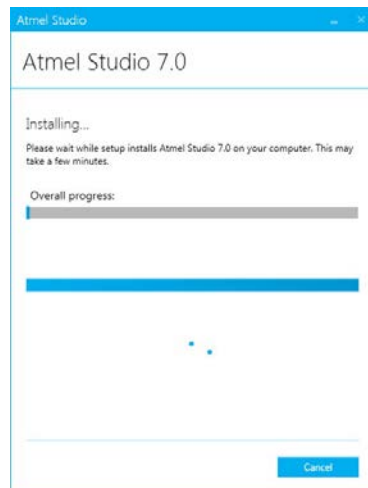
6. ในขณะที่ทำการติดตั้ง โปรแกรมจะแสดงผลดังรูป



7. เมื่อโปรแกรมขอให้ Restart ระบบดังรูป ให้คลิก Restart Now



8. หลังจากที่เรา restart เครื่องพีซีก็จะทำการติดตั้งโปรแกรมต่อไปอีกดังรูป

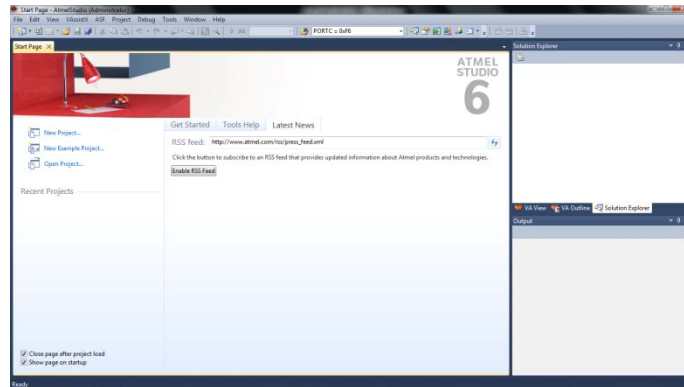


9. เมื่อการติดตั้งสมบูรณ์ดังรูปให้คลิก Close

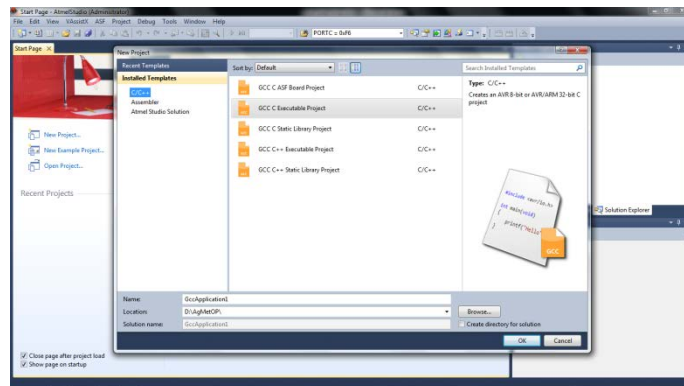


การแปลงโปรแกรมการทำงานของไมโครคอนโทรลเลอร์จากภาษาซีให้เป็น hex ไฟล์

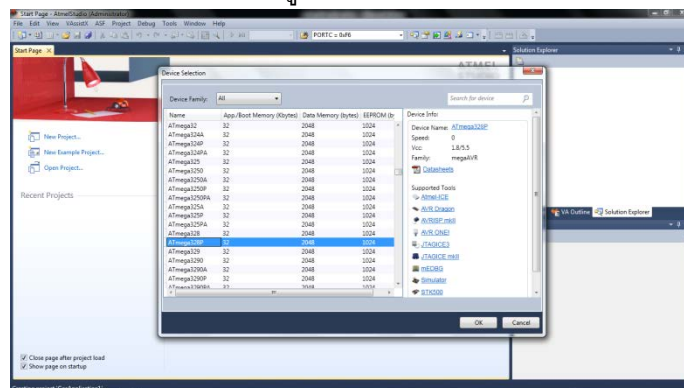
1. เมื่อคลิก Atmel Studio (AS) เพื่อเริ่มการทำงานจะเห็นดังภาพ (ภาพที่แสดงเป็นภาพของ Atmel Studio version 6.2 ซึ่งอาจแตกต่างไปจาก version 7)



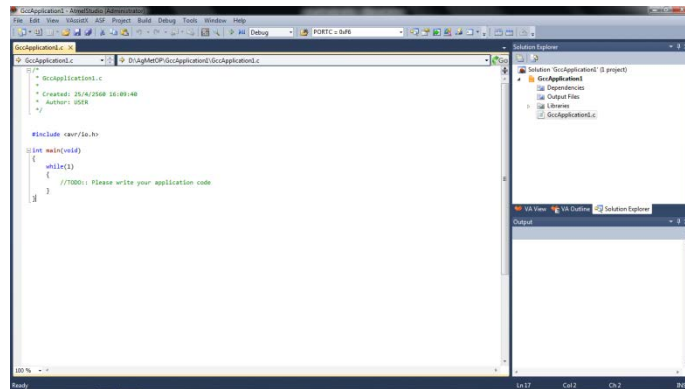
2. เมื่อคลิกที่ New Project ที่ด้านซ้ายมือของหน้าต่าง AS จะแสดงหน้าต่างเพื่อใช้กำหนดลักษณะของงาน ให้เลือก GCC C Executable project ดังรูปและกำหนดชื่อของ project และ โฟลเดอร์ ลงในช่อง Name และ Location จากนั้นคลิก OK



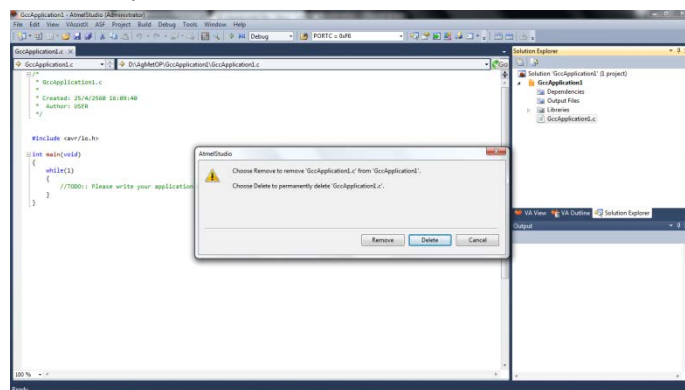
3. จากหน้าต่างที่ AS แสดงดังรูป ให้เลื่อนแถบเลือกลงและเลือก Atmega328P จากนั้นคลิก OK



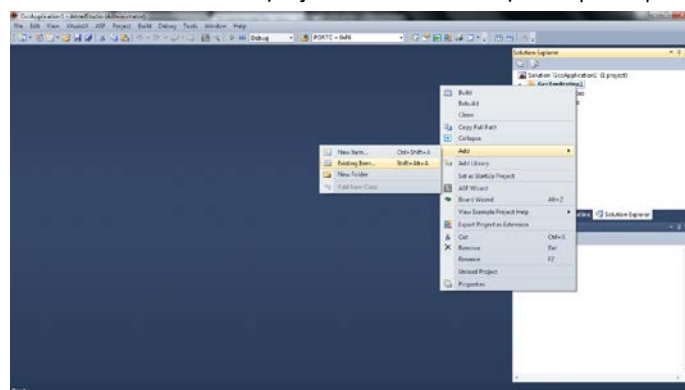
4. AS จะสร้างไฟล์ของซอร์สโค้ดเพื่อใช้เริ่มงานโดยตั้งชื่อตามชื่อของ project



5. แต่ในกรณีนี้จะใช้ซอร์สโค้ดที่ได้เขียนไว้แล้ว ดังนั้นให้คลิกขวาที่ชื่อของไฟล์ดังกล่าวในหน้าต่าง Solution Explorer และเลือก delete

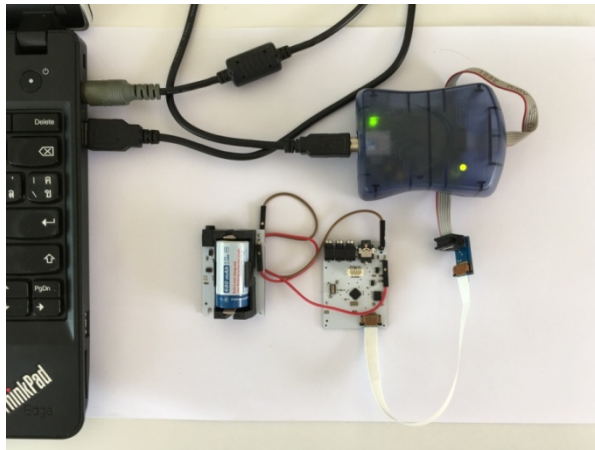


6. จากนั้นคลิกขวาที่ชื่อของ project ใน Solution Explorer | Add | Existing Item

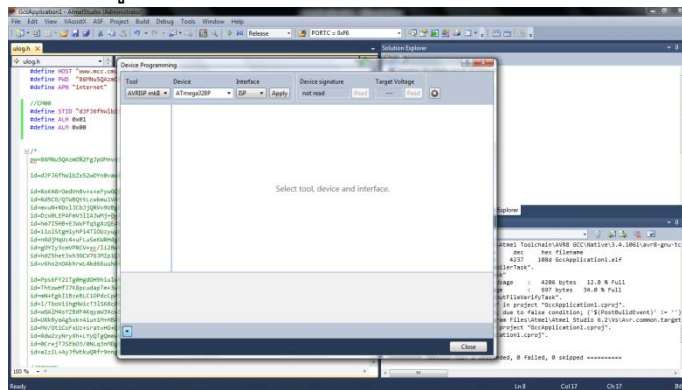


7. จาก explorer ให้เลือก 18 ไฟล์จากโฟลเดอร์ Station (คือ client.c, adc.c, adc.h, uart.c, uart.h, udfm.c, udfm.h, uint.c, ulog.c, ulog.h, urtc.h, usht.c, usht.h, utcn.c, wrtc.c, wsetrtc_HTP.c, wtcp.c, และ wtcp.h) แล้วคลิกปุ่ม Add

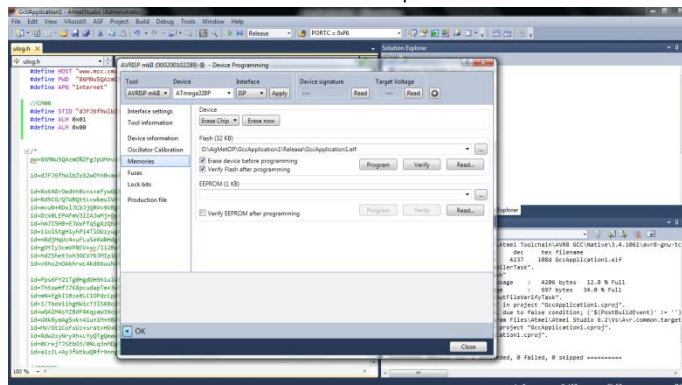




2. บนแถบเมนูของ AS ให้คลิก Tool | Device Programming



3. ในช่อง Device Programming ให้คลิกปุ่ม Apply แล้วคลิกเลือก Memories และคลิกปุ่ม Program



- เมื่อ AS ทำการอัปโหลดโปรแกรมจะแสดงผลของการทำงานคือ Erasing device...OK, Programming Flash...OK และ Verifying Flash...OK ไว้ที่ด้านล่างของหน้าต่าง หากได้ผลลัพธ์ดังแสดงในภาพ การอัปโหลดเสร็จสมบูรณ์ ให้ถอด วงจรไมโครคอนโทรลเลอร์ออกจาก programmer

เมื่อทำการอัปเดตโปรแกรมแล้วให้ทำการตรวจสอบแผงวงจรของไมโครคอนโทรเลอร์โดยใช้คำสั่ง AT ทั้ง 11 ขั้นตอนอีกครั้ง หากวงจรของไมโครคอนโทรเลอร์ยังไม่ทำงานแสดงว่า ส่วนที่เสียหายนั้นอาจเกิดจากตัวไมโครคอนโทรเลอร์เองหรือ นาฬิกา หรือส่วนอื่นๆ

การตรวจสอบไอซีของไมโครคอนโทรเลอร์และนาฬิกาเวลา

การตรวจสอบไอซีของไมโครคอนโทรเลอร์

1. สร้าง Project ใหม่ของ Atmel Studio และ ดำเนินการตามขั้นตอนที่ได้กล่าวในรายละเอียดไว้แล้ว ในบทที่ 2 ในหัวข้อการแปลงโปรแกรมการทำงานของไมโครคอนโทรเลอร์จากภาษาซี ให้เป็น hex ไฟล์ โดยเลือกไฟล์ชื่อ ctrl.c จากโฟลเดอร์ uCTRL
2. อัปโหลด hex ไฟล์ที่ได้เข้าสู่วงจรไมโครคอนโทรเลอร์ ดังที่ได้กล่าวในรายละเอียดไว้แล้วในบทที่ 2 ในหัวข้อการอัปโหลด hex ไฟล์เข้าสู่ไมโครคอนโทรเลอร์
3. จากนั้นให้เชื่อมต่อ วงจรไมโครคอนโทรเลอร์เข้ากับ USB ของเครื่องพีซี PC โดยใช้อุปกรณ์แปลงสัญญาณ TTL-USB ดังที่ได้กล่าวในรายละเอียดไว้แล้วในบทที่ 1
4. เปิดโปรแกรม terminal จ่ายกระแสไฟฟ้าให้กับวงจรไมโครคอนโทรเลอร์ หากไมโครคอนโทรเลอร์ ไอซีไม่เสียหาย จะเห็นข้อความปรากฏขึ้นบนหน้าต่าง terminal ดังนี้

16:21:38.568> Testing uCTRL

16:21:38.568>

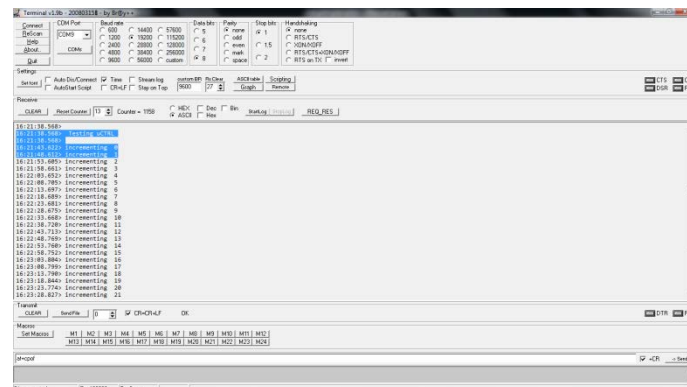
16:21:43.622> incrementing 0

16:21:48.612> incrementing 1

16:21:53.605> incrementing 2

16:21:58.661> incrementing 3

...



การตรวจสอบไอซีของนาฬิกา

1. สร้าง Project ใหม่ของ Atmel Studio และ ดำเนินการตามขั้นตอนที่ได้กล่าวในรายละเอียดไว้แล้วในบทที่ 2 ในหัวข้อการแปลงโปรแกรมการทำงานของไมโครคอนโทรเลอร์จากภาษาซี ให้เป็น hex ไฟล์ โดยเลือกไฟล์ชื่อ setRV3029c.c, RTC_Param.c, rtc.c, uart.c, uart.h, interrupt.c, และ urtc.h จากโฟลเดอร์ RTC
2. อัปโหลด hex ไฟล์ที่ได้เข้าสู่วงจรไมโครคอนโทรเลอร์ ดังที่ได้กล่าวในรายละเอียดไว้แล้วในบทที่ 2 ในหัวข้อการอัปโหลด hex ไฟล์เข้าสู่ไมโครคอนโทรเลอร์
3. จากนั้นให้เชื่อมต่อ วงจรไมโครคอนโทรเลอร์เข้ากับ USB ของเครื่องพีซี PC โดยใช้อุปกรณ์แปลงสัญญาณ TTL-USB ดังที่ได้กล่าวในรายละเอียดไว้แล้วในบทที่ 1

4. เปิดโปรแกรม terminal จ่ายกระแสไฟฟ้าให้กับวงจรไมโครคอนโทรลเลอร์ หากไอซีนาฬิกาไม่เสียหาย จะเห็นข้อความปรากฏขึ้นบนหน้าต่าง terminal และมีการเปลี่ยนแปลงของเวลาทุกๆ นาที ดังนี้

9:24:08.830> Setting date and time

9:24:08.830>

9:24:31.730>

9:25:31.699> YM DH MS: 1703 2212 0600

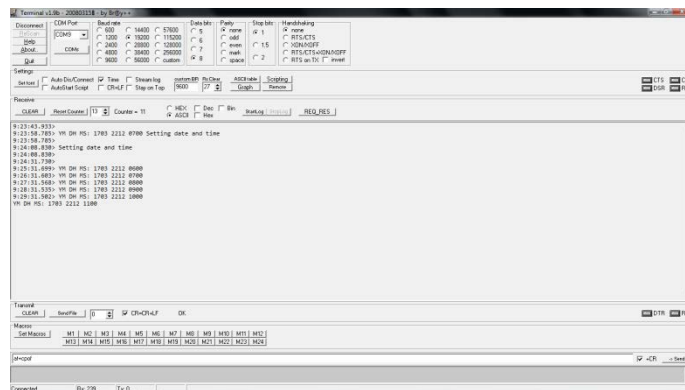
9:26:31.603> YM DH MS: 1703 2212 0700

9:27:31.568> YM DH MS: 1703 2212 0800

9:28:31.535> YM DH MS: 1703 2212 0900

YM DH MS: 1703 2212 1000

...



Development of wireless weather station for precision agriculture

Krirk Pannangpetch^{a,*}, Thewin Kaeomuangmoon^b

^a Department of Plant Science and Agricultural Resource Management, Faculty of Agriculture, Khon Kaen University, Khon Kaen 40002, Thailand.

^b Multiple Cropping Center, Faculty of Agriculture, Chiangmai University, Chiangmai 50200, Thailand.

Abstract

Needs to maximize input efficiency and yield while minimize impact on environment have driven food production system toward information based precision agriculture, in which near real time weather information is one of the fundamental components. Therefore, wireless automatic weather station was developed to enable the construction of dense weather network and transmission of near real time weather data from the field via 3G-internet network to a web server.

The wireless weather station developed in this work is accurately measured solar radiation, air temperature, air relative humidity, wind speed, and rainfall. Transmission of hourly weather data to web application through 3G-Internet is reliable, provided that the 3G signal strength is better than -90 dBm.

Key words: wireless weather station, precision agriculture, food security.

Introduction

Needs to maximize input efficiency and yield while minimize impact on environment have driven food production system toward information based agriculture, in which near real time weather information from dense weather station network is one of the fundamental components (Lowenberg-DeBoer, 2015; Mello and Treinish, 2015; Herring, 2001). Long distant transmission of weather data can be made via cellular-Internet network (Liqiang et al 2011; Shining et al 2011; Kassim et al. 2014; Chaudhary et al, 2011; Sakthipriya. 2014; and Baggio, 2015) while protocol such as Zigbee, WiFi or Bluetooth can be used for short distant transmission (Rehman et al, 2014; Camilli et al, 2007; Keshtgary and Deljoo, 2012; Wang et al, 2006; and Garcia-Sanchez et al. 2011). High density installation of weather station is essential but costly. In order to meet such a requirement, a cost effective wireless automatic weather station was developed to allow transmission of near real time weather data from the field via 3G-internet network to a web server, where the data can be freely accessed.

* Corresponding author at: Department of Plant Science and Agricultural Resource Management, Faculty of Agriculture, Khon Kaen University, Khon Kaen 40002, Thailand. E-mail address: krirk@kku.ac.th

Design of the hardware

Data Processing Unit

High performance low-power RISC 8 bit microcontroller with 32 KBytes of In-System Self-Programmable Flash program memory, 1KBytes EEPROM, and 2K Bytes Internal SRAM. Built-in peripherals also include three Timer/Counters, internal and external interrupts, a serial programmable USART, 2-wire Serial Interface, an SPI serial port, and 6-channel 10-bit ADC.

RF module

Single band HSDPA 2100MHz System on Chip (SoC) transceiver designed for M2M wireless application which also includes UART interface and embedded TCP/IP protocol.

External flash memory

2Mbytes of 512/528 per page, with minimum of 100000 Program/Erase cycles and data retention of 20 years

Real Time Clock

Temperature Compensation is calibrated in the factory and yields time accuracy of $\pm 6\text{ppm}$ from -40°C to $+85^\circ\text{C}$ and $\pm 8\text{ppm}$ from -40°C to $+125^\circ\text{C}$.

Power Supply

Monocrystalline high efficiency solar cells, rechargeable lithium ion battery, and an integrated Li-ion linear charger.

Sensors

Solarimeter is based on photovoltaic IC with built-in OpAmp, Temperature and humidity based on single chip module providing a calibrated digital output, anemometer and raingauge based on CMOS hall ICs.

Design and integration of the system hardware is illustrated as schematic in Figure 1.

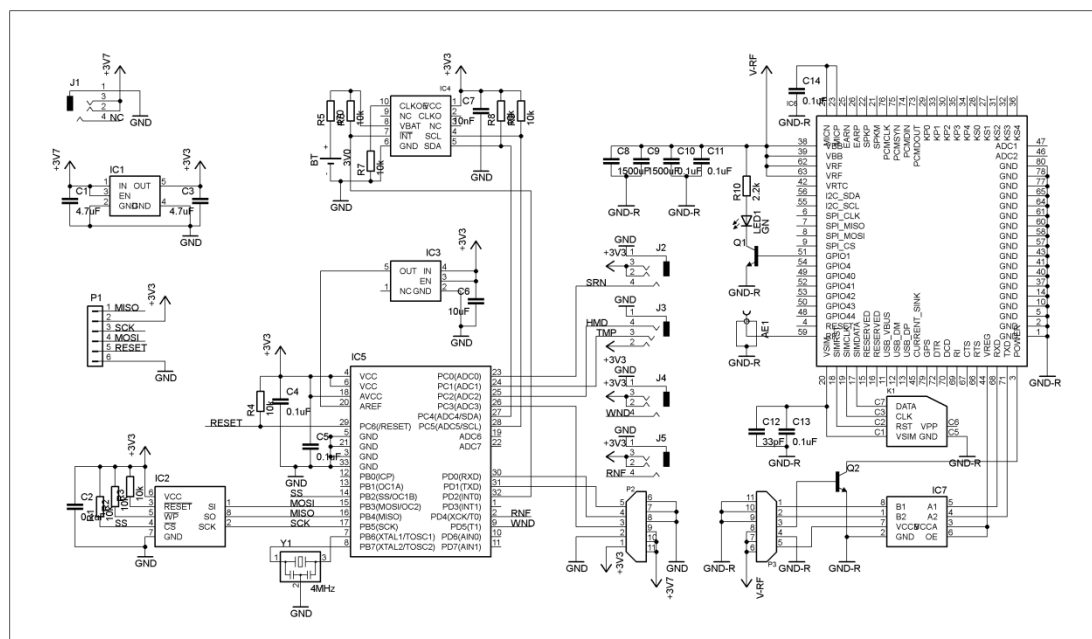


Figure 1 Schematic of the design hardware

System software

Operation system of the data logger was written in C and the algorithm of the code is as follow. The microcontroller is initialized into sleeping state to conserved energy, and would only be woken up by real time clock at the pre-set time interval to collect data from sensors, and then write the data to the external flash memory. If it is time to send data to remote server, then it would start RF module to send data using TCP/IC protocol, and go back to sleep. Flow of the operation system is displayed as flow chart in Figure 2.

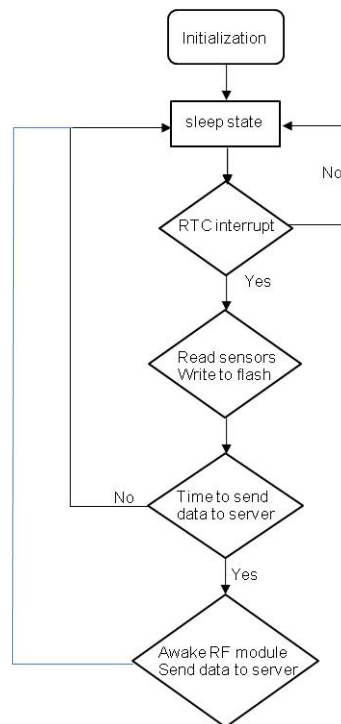


Figure 2 Flow of system operation

Experimental results

Accuracy

Data collected by the developed weather station (uMet) were compared with those collected by Campbell Scientific Inc weather station which comprised Campbell CR10X data logger, Licor Li200 pyranometer, Vaisala HMP45 temperature and Humidity sensor, R.M. Young 03101 anemometer, and Sierra-Misco Model 2502 rain gauge.

The results showed that measurements made by the developed low cost weather station were almost identical to those by Campbell Scientific Inc system (Figure 3 to 7).

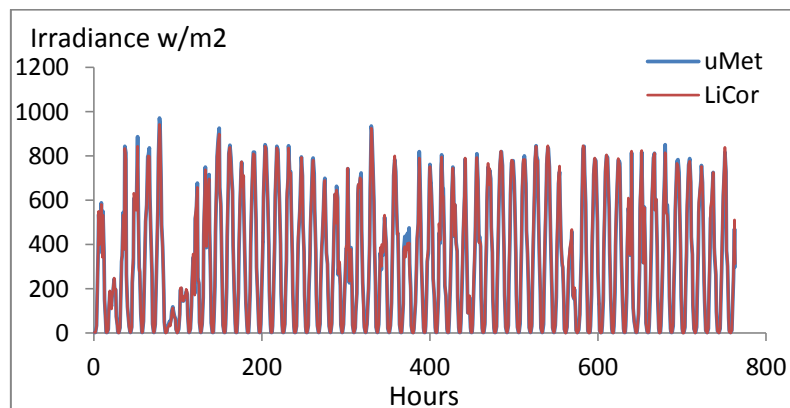


Figure 3 Comparison of irradiance as measured by the developed weather station (uMet) and that by Campbell weather station (LiCor)

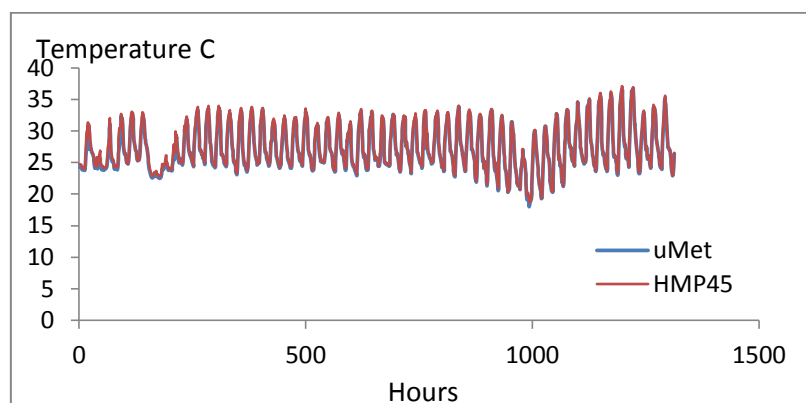


Figure 4 Comparison of air temperature as measured by the developed weather station (uMet) and that by Campbell weather station (HMP45)

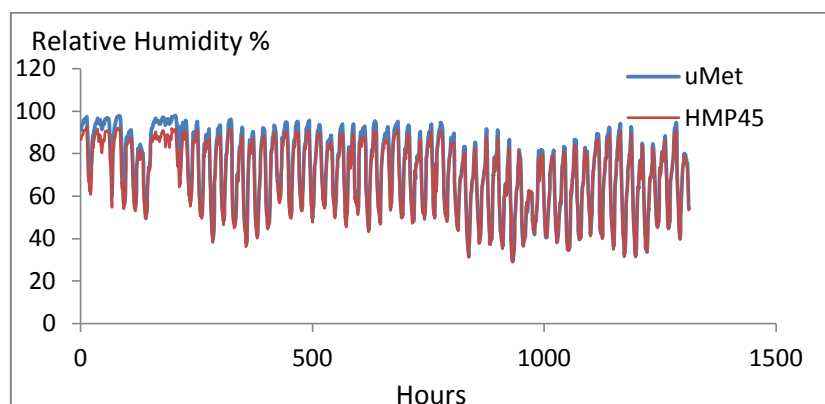


Figure 5 Comparison of air relative humidity as measured by the developed weather station (uMet) and that by Campbell weather station (HMP45)

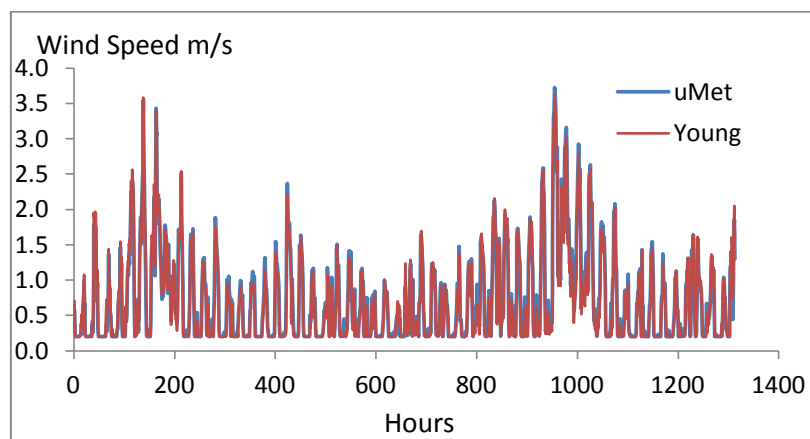


Figure 6 Comparison of wind speed as measured by the developed weather station (uMet) and that by Campbell weather station (Young)

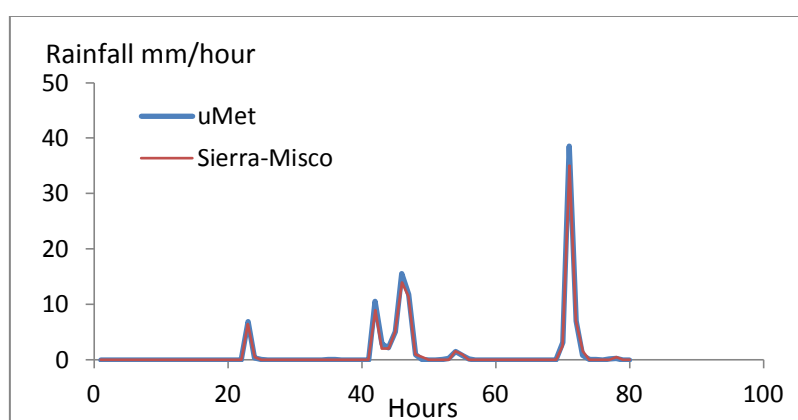


Figure 7 Comparison of rainfall as measured by the developed weather station (uMet) and that by Campbell weather station (Sierra-Misco)

Similarity between the data from the developed weather station and those from Campbell Scientific Inc was further quantified by means of simple linear regression parameters: slope and intercept, and deviation from linearity. From Figure 8 it can be seen that measured values obtained from the developed weather station are linearly related to those values measured by Campbell Scientific Inc weather station. The slope of the relationship is very close to 1 with the intercept close to 0. Deviation from linear line is very low and in all cases it was less than 0.1 (r^2 is greater than 0.9). The only exception is that of wind speed which revealed none linearity at low wind speed. This is contributed to the design and property of R.M. Young anemometer 03101 that has the offset value of 0.2 m s^{-1} —wind speed less than 0.2 m s^{-1} will be recorded as 0.2 m s^{-1} .

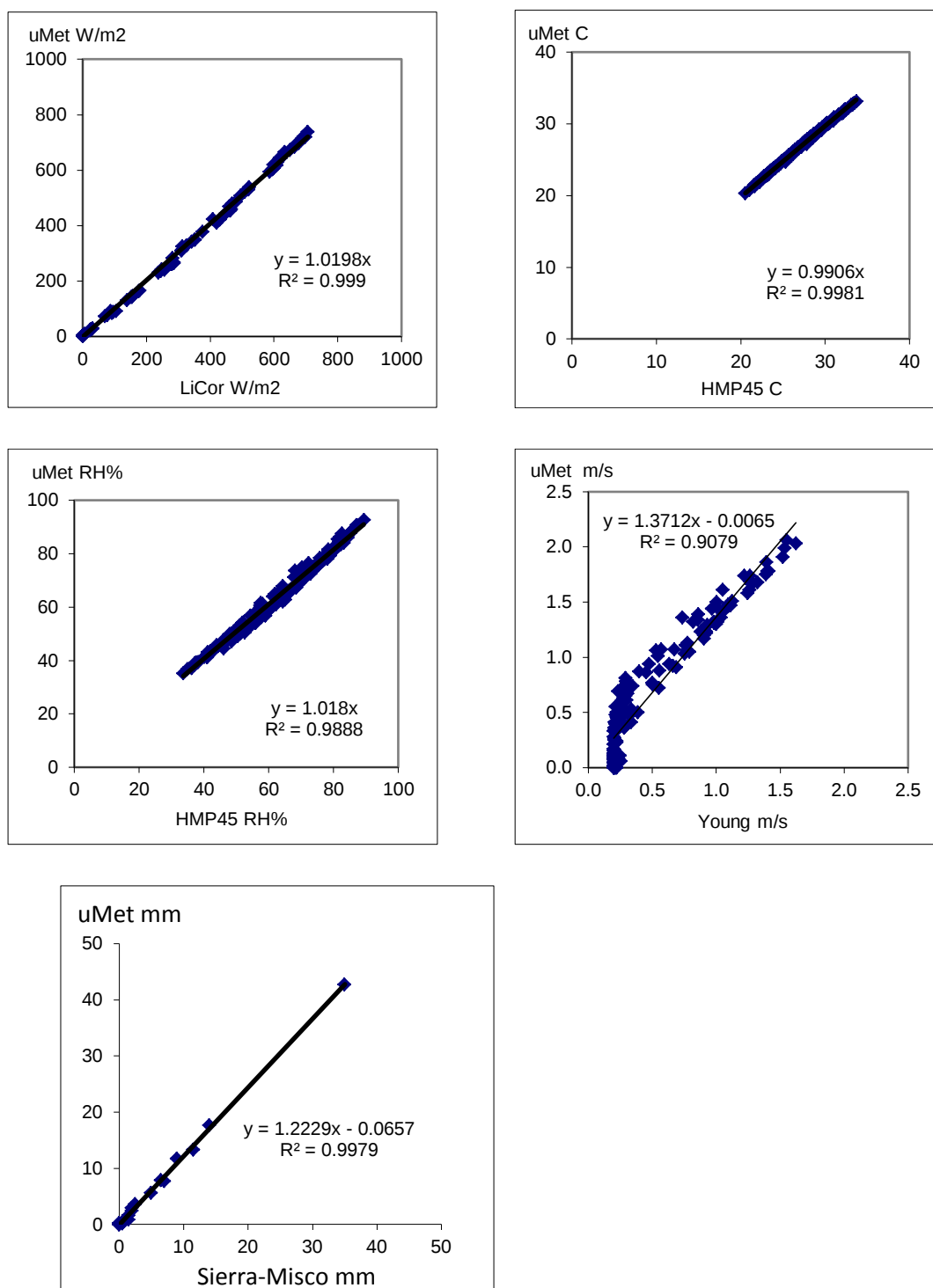


Figure 8 Similarity of weather variables recorded by the developed weather station (uMet) and those by Campbell weather station as quantified by means of simple linear regression.

Stability and reliability

Thirteen stations were installed in the field in Thailand: nine in orange groves in Chai Prakan, Fang and Mae Ai district, ChiangMai province: one at Mae Hia Agricultural Research Center, Faculty of Agriculture, ChiangMai University: one in Mueang district, Phayao Province: one in Suan Kahakarkaset, Lat Lum Kaeo distric, Phatum thani Province: and one in vineyard, Muak Lek district, Saraburi province (Figure 9). 3G signal strength in these locations ranged from -75 dBm to less than -95 dBm. Testing was conducted from March 2017 to September 2017.

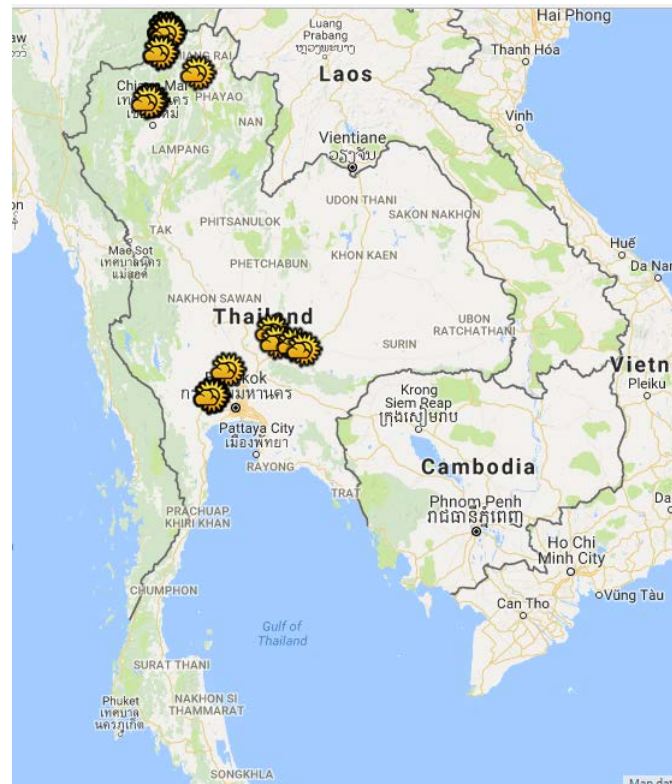


Figure 9 Locations in Thailand where thirteen weather stations were installed and tested under field conditions.

Results showed that eleven weather stations installed in area where 3G signal strength was better than -95 dBm consistently and correctly sent hourly weather data from the field to web server, while the other two which were installed in area where signal strength lower than -95dBm intermittently fail to transmit data. Therefore to ensure reliable and stability of data transmission, especially during heavy rain, the station should be installed in area where 3G signal strength is better than -90 dBm.

Conclusion

The wireless weather station developed in this work is accurately measured solar radiation, air temperature, air relative humidity, wind speed, and rainfall. Data Transmission to web application through 3G-Internet is stable and reliable, provided

that the 3G signal strength is better than -90 dBm. Thus it can be used to construct dense weather station network and provide near real time weather information.

Acknowledgement

This work was supported by Thailand Research Fund **TRF** (RDG53O0009, RDG 5650128, and SRI5920205)

References

- Baggio, A. 2015. Wireless sensor networks in precision agriculture. Delft University of Technology, The Netherlands. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.120.46&rep=rep1&type=pdf>.
- Camilli, A., C. E. Cugnasca, A. M. Saraiva, A. R. Hirakawa, P.L.P. Correa. 2007. From wireless sensors to field mapping: Anatomy of an application for precision agriculture. *Computers and Electronics in Agriculture*. 58, 25–36.
- Chaudhary, D.D, S.P.Nayse, L.M.Waghmare. 2011. Application of wireless sensor network for greenhouse parameter control in precision agriculture. *International Journal of Wireless & Mobile Networks (IJWMN)*.3(1), 140-149.
- Garcia-Sanchez, A, F. Garcia-Sanchez, J. Garcia-Haro. 2011. Wireless sensor network deployment for integrating video-surveillance and data-monitoring in precision agriculture over distributed crops. *Computers and Electronics in Agriculture*. 75, 288–303.
- Herring, D. 2001. Precision farming. <http://earthobservatory.nasa.gov/Features/PrecisionFarming/>.
- Kassim, M.R.M., I. Mat, and A.N. Harun. 2014. Wireless Sensor Network in Precision Agriculture Application. 1-5. In *International conference on Computer, Information and Telecommunication Systems (CITS)*, IEEE Publication, ISBN 978-1-4799-4384-5.
- Keshtgary, M. and A Deljoo. 2012. An efficient wireless sensor network for precision agriculture. *Canadian Journal on Multimedia and Wireless Networks*. 3(1), 1-5.
- Liqiang, Z., Y. Shouyi, L. Leibo, Z. Zhen, W. Shaojun. 2011. A Crop Monitoring System Based on Wireless Sensor Network. *Procedia Environmental Sciences*.11, 558–565.
- Lowenberg-DeBoer, J. 2015. The Precision Agriculture Revolution: Making the Modern Farmer. *Forreignaffairs*, ESSAY May/June 2015 Issue, United States Economic Development. <https://www.foreignaffairs.com/articles/united-states/2015-04-20/precision-agriculture-revolution>.
- Mello, U and L.Treinish. 2015. Precision agriculture using predictive weather analytics to feed future generations. http://www.research.ibm.com/articles/precision_agriculture_shtml.
- Rehman, A.,A. Z. Abbasi, N. Islam, Z. Shaikh. 2014. A review of wireless sensors and networks' applications in agriculture. *Computer Standards & Interfaces*. 36, 263–270.
- Sakthipriya, N. 2014. An Effective Method for Crop Monitoring Using Wireless Sensor Network. *Middle-East Journal of Scientific Research*. 20(9), 1127-1132.

- Shining, L., C. Jin, and L. Zhigang. 2011. Wireless Sensor Network for Precise Agriculture Monitoring, 307-310. In International Conference on Intelligent Computation Technology and Automation (ICICTA). Vol. 1, IEEE Publication, ISBN 978-1-61284-289-9.
- Wang, N., N. Zhang, and M. Wang. 2006. Wireless sensors in agriculture and food industry—Recent development and future perspective. *Computers and Electronics in Agriculture*.50, 1–14.

ตารางเปรียบเทียบวัตถุประสงค์ กิจกรรมที่วางแผนไว้ และกิจกรรมที่ ดำเนินการมาและผลที่ได้รับ

วัตถุประสงค์	กิจกรรมตามแผน	ผลที่คาดว่าจะได้รับ ตามแผน	ผลการ ดำเนินงาน	หมายเหตุ
พัฒนาศถานีตรวจวัด อากาศแบบไร้สายให้ ส่งข้อมูลสภาพอากาศ จากภาคสนาม ผ่าน เครือข่าย 3G และ internet ไปยัง web server	1 ออกแบบและสร้าง แผงวงจรอิเล็กทรอนิกส์ ของ 3 จี โมดูล และ ปรับเปลี่ยนวงจร อิเล็กทรอนิกส์ และ ชุดคำสั่งของเครื่อง บันทึกข้อมูล ให้ สามารถส่งข้อมูลผ่าน 3 จี และ อินเทอร์เน็ต ไป ยังเว็บเซิร์ฟเวอร์	ได้ต้นแบบ 3จี โมดูล และเครื่องบันทึกข้อมูล (AgDAPAw 1.0) ที่ สามารถส่งข้อมูลสภาพ อากาศจากภาคสนาม ผ่านเครือข่าย 3 จี และ อินเทอร์เน็ต ไปยังเว็บ เซิร์ฟเวอร์	เป็นไปตามแผน	
	2 สร้างเซ็นเซอร์ วัด สภาพอากาศ 3 จีโมดูล และเครื่องบันทึกข้อมูล จำนวน 20 ชุด พร้อม เทียบค่ากับอุปกรณ์ ของบริษัท Campbell Scientific Inc.	ได้สถานีวิจัยตรวจวัดอากาศ แบบไร้สาย (AgDAPAw 1.0) และติดตั้งสถานี ดังกล่าวในระบบเกษตร แม่นยำ	เป็นไปตามแผน	
	3 ทดสอบสถานี ตรวจวัดอากาศแบบไร้ สายที่ได้จากข้อ 2 ใน สภาพแปลง เพื่อทำการ ปรับปรุงหรือแก้ไข ข้อผิดพลาด	มีอุปกรณ์ AgDAPAw 1.0 และข้อมูล รายละเอียดสูงพร้อมใช้ งานกับการเกษตรแบบ แม่นยำ	เป็นไปตามแผน	