Reduction of Agricultural Water Consumption through the Utilization of a Wireless Sensor Array

Ankur Raghavan
Bethpage High School
Bethpage, New York
January 13, 2023

All figures and photographs were taken by the student unless otherwise stated

Abstract

With over 70% of all freshwater worldwide used for agriculture, it is important that none of it is wasted. However, overwatering is a common occurrence on agricultural farms. If farmers do not properly adjust their irrigation timings to reflect real-world situations, they can end up using more water than they need to, costing $300 billion. Currently, less than 12% of all farms use soil moisture sensors in any capacity. To help reduce the amount of water wasted in agriculture, a wireless soil moisture sensor array was prototyped. This low-cost network can transmit sensor readings over 400 meters to a central base station, reducing the number of expensive base stations needed. These readings were then used to automatically turn on and off sprinklers and the farmers were able to view them. This prototype has an expected battery life of over 6 years, thereby it needs little maintenance. This product has the potential to reduce the hundreds of billions of dollars wasted annually in excess water. This technology costs $33.38 for a sensor, $32.75 for a base station, and $16.95 in one-time costs. An average farm would need $515 to fully outfit their farm. Compared to current technology, this prototype is 20 times cheaper. It also has a larger range than options available on the market now.

Introduction

The world agricultural production needs to double to keep up with population growth. As 70% of all freshwater worldwide is used for agriculture, farms need to be more efficient in managing their water resources while increasing their productivity (Swain et al. 2021). Currently, 70% of farmers' time is spent monitoring the crops and their conditions (Swain et al. 2021). Use of automated sensors to monitor soil conditions could reduce the amount of time. However, current sensor technology is not adapted for bigger farms since many still use range-limited devices (Swain et al. 2021). Long Range Wide Access Network (LoRaWAN) is a new technology that has been made to allow long-range and low-power communications. This new technology allows the sensors to be spread out, and on battery power, but still communicate and last long enough for the entire growing season.

LoRa is a standard of communication on the 915 MHz license-exempt band. This band is reserved for industrial, scientific, or medical uses and does not require a license to transmit on so it will be effective for an agricultural use. Additionally, LoRa should only be used if there are many devices and they are far away from each other since it comes with significant throughput restrictions (Augustin et al. 2016). Another standard that could be used is the Institute of Electrical and Electronics Engineers (IEEE) 802.11ah, a low-power Wi-Fi network. It also runs on the 915 ISM band, so you do not need a license to use it. IEEE 802.11ah has an indoor range of 50 meters, but that comes with a cost of a significantly slower speed at less than 1 Mbps. This standard has a much higher throughput, but that comes with range limitations which is why it was not chosen for this project (Šljivo et al. 2018). Restricted Access Window (RAW) schemes can be implemented to reduce power draw, since the device will only be awake and using power during specific time slots, to receive and send transmissions (Wang et al. 2017). While there are ways to slightly improve the speed and range of IEEE 802.11ah, LoRa has a significantly greater range so it is more effective for this scenario. The sensors will have very little data to send to the base station so LoRa's range is more important than 802.11ah's speed; however, if you wanted images of the crops or live video, 802.11ah would be much better since it has enough speed and bandwidth to have many live videos broadcasted to the same access point (Šljivo et al. 2018).

LoRa also has great battery life and since it is only powered up when it is measuring sensor data and transmitting it, it can last a full growing season on the same batteries (Fisher et al., 2018). Additionally, you could use solar panels mounted above the plants to have the sensors

recharge and not worry about them losing battery (John et al., 2018). Some sensors that are often used are temperature, humidity, and soil moisture sensors to determine whether the optimal conditions are met for the plant.

Soil moisture sensors are critical because they can help prevent overwatering. Overwatering occurs when more water is used without gaining crop yield. 32% of the annual water volume used to irrigate maize could have been saved without affecting the yield of crops (Grassini et al., 2011). Since maize currently uses over 7 trillion gallons of water per year, 2.3 trillion gallons of water could be saved by farmers ("USDA"). This would save over $300 billion (Wichelns, 2010).

The goal of this project is to create a lower-cost alternative to current soil monitoring systems, which could work over larger distances and for long periods. Additionally, this system should be easily expandable without altering already installed sensors. The system should have a live readout of the current conditions such that the farmers do not have to go out to the field to inspect the conditions.

Methodology

Engineering Goals

The engineering goals are to create a low-cost alternative for water sensing, have effective communication over at least 100 meters, and three month battery life.

Subjects/Materials/Instruments

- 2 Moteino Wireless Transceivers
- 1 USB FTDI programming board
- 18 Jumper Wires
- 1 Moisture Sensor
- 1 Battery
- 1 Solid State Relay
- 2 Prototyping Boards
- Arduino IDE

Protocols

1. Wired the receiving transceiver

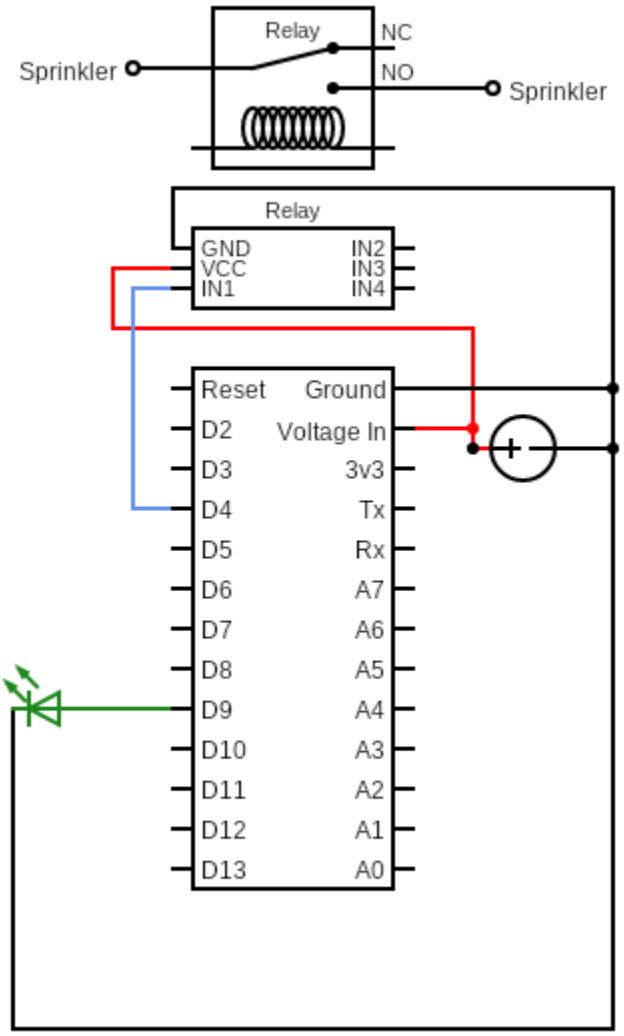

Figure 1. Wiring Diagram of the Receiving Transceiver

a. Soldered the included male headers to a transceiver such that the main chip is on the opposite side of the headers
b. Three wires were connected to the ground pin and two to the Voltage In pin.
    i. One set of these wires was connected to the relay module
    ii. Another set was used as the power input and connected to the battery or other power source
    iii. The final ground wire was connected to the led
c. The LED was also connected to the D9 pin
    i. This pin was used because it also controls the onboard LED
d. The Relay module was also connected to the D4 pin

2. Wired the sensor transceiver

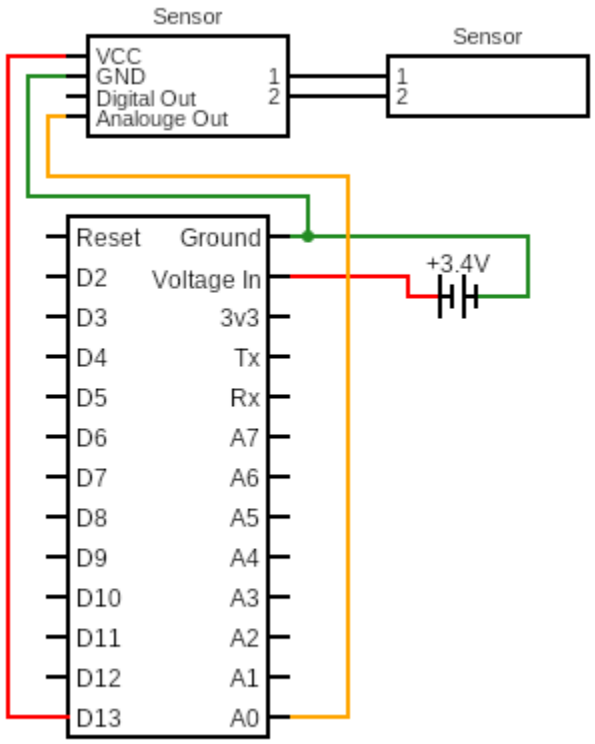

Figure 2: Wiring diagram of the sensor transceiver

   a. Soldered the included male headers to a transceiver such that the main chip is on the opposite side of the headers
   b. 2 wires were connected to each of the ground and voltage in pins
      i. One set of these wires was connected to the soil moisture sensor
      ii. Another set was used as the power input and connected to the battery
   c. Another wire was connected from the analog out pin on the sensor to the A0 pin on the transceiver
3. Uploaded code to each transceiver
4. Created housing for each system
   a. Modeled the housing in Fusion 360
   b. 3D printed it in a bright color for visibility
5. Tested range of the sensor
   a. Temporarily decreased the time between each transmission
   b. Placed the transmitting sensor in a stable position and logged the GPS position

c. Slowly walked away from the sensor while holding the base station and watching the LED
d. Once the LED stops blinking for 30 seconds log the GPS position
e. Find the distance between the two spots and log it in a spreadsheet
f. Repeat for multiple places and find the average range

Code Description

The code for controlling the sensors and the sprinkler relay was written in the Arduino Integrated Development Environment (IDE). This coding language is very similar to both C and C++ and has features from both.

The transceivers' code uses the Low Power Labs AVR Board Library and the RFM69 Library. Two different programs were created. The first one controls the base station and sprinkler. The program is always listening for any incoming packages of information. If the project ID matches, it will send an acknowledgment. It will also blink the onboard and external LED to display activity. Then it logs the moisture sensor's data, sensor id, and current time. If there is sufficient soil moisture from that sensor, it will turn off the sprinkler relay for the corresponding section of the field. The user can tune this threshold. See Appendix A for the full code.

The second program controls the individual sensors. This program measures the sensor data at a specified interval (every hour). It then sends the sensor data, along with two unique ids: one for this project to distinguish it from other devices that are used in the farm, and the other unique to each sensor to identify its location. After the data is sent, the program waits for an acknowledgment. Once the acknowledgment is received, the device goes to sleep for the specified interval, to reduce battery usage. See Appendix B for the full code.

Results

Figures 3-5 depict the final base station and Figures 6-8 depict a final sensor station.

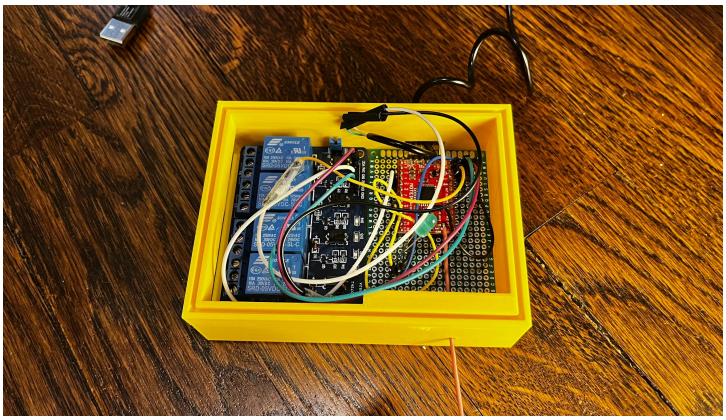

Figure 3: Top view of the Base Station without the lid on

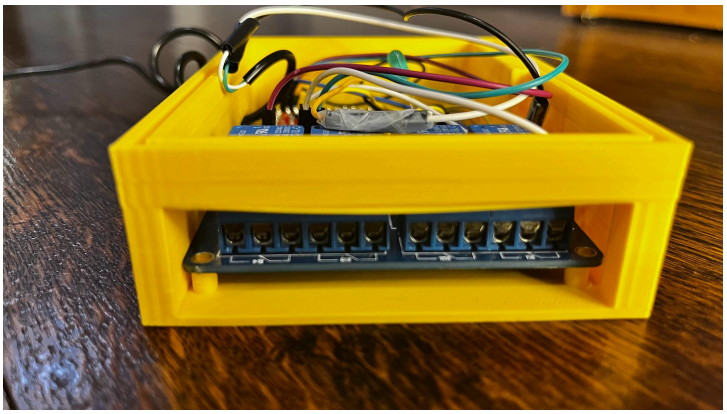

Figure 4: Side view of the opening for access to the relay

Figure 5: Side view of the closed base station

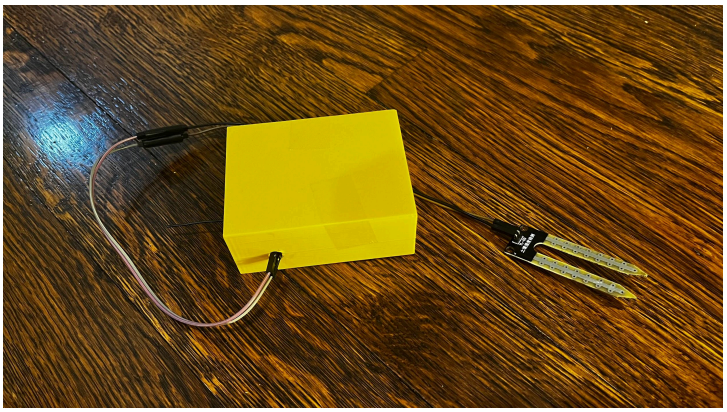

Figure 6: Top view of the closed sensor/transmitter

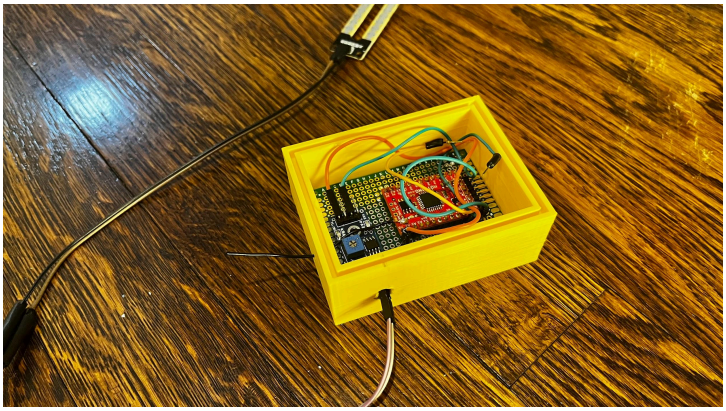

Figure 7: Top view of the sensor/transmitter without the lid

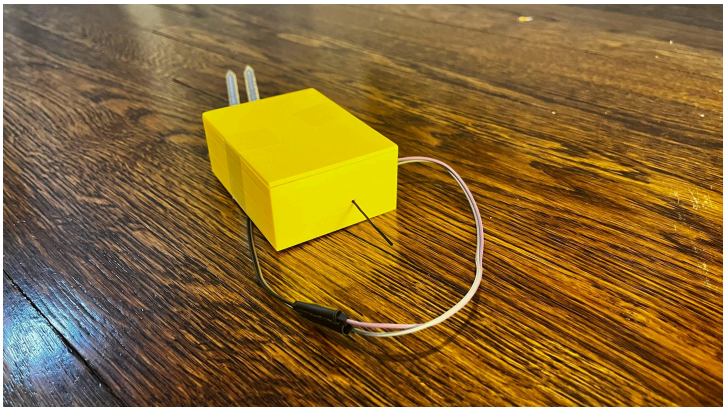

Figure 8: Side view of the closed sensor/transmitter

The transceiver, relay module, and activity LED are visible in Figure 3. In the side view of the base station in Figure 4, the relay module is accessible through a hole. This was designed such that any sprinkler cables can be easily connected to the relay. Figure 5 shows the protective housing for the electronics. This was 3D printed such that it matches the product exactly. The USB power cable is passed through a hole in the casing so it can be easily connected to a power source.

Figure 6 shows the sensor/transmitter module in its protective housing. As for the base station, the housing was 3D printed to make it fit perfectly. The housing contains a hole for the sensor wires to leave as shown in figure 6. Figure 7 shows the transmitter and sensor logic board inside the housing. Figure 8 shows the hole that the antenna uses to communicate with the base station. The power cable exits through a hole in the housing and is attached to the battery separately from the main housing.

Table 1: Range data of the transceivers

Trial Num	Range (miles)	Range (meters)
1	0.25	402.336
2	0.22	354.05568
3	0.26	418.42944
4	0.25	402.336
5	0.24	386.24256
6	0.27	434.52288
Average	0.2483333333	399.65376

Table 1 shows the ranges of communication found in six different trials conducted in my town. A line of sight between the transmitter and receiver was ensured. The average distance was .248 miles or almost 400 meters.

Discussion

Range

Using the range of 400 meters found, any sensor within an area of over 5 hundred thousand square meters can communicate with a single base station. How many farmers utilize their land makes this solution more valuable. Many farmers use a 400-meter center spinning sprinkler to water their crops (G. Evans, 2001). Since this prototype can reach a range of about 400 meters, a base station could be placed in the center of each circle and can effectively service that entire field. Additionally, this range means that many sensors can be placed within the area of a single base station. The more sensors that are used, the more detailed a picture the farmer can use to plan watering schedules.

Durability

Due to the long-term usage of this prototype, it needs to be durable and withstand outdoor conditions for extended periods. To increase the durability of this prototype, it was placed inside a watertight container. Additionally, due to this solution being used outdoors, it needed to be able to withstand extreme temperatures. Every component is rated to operate normally up to 100 degrees Celsius and down to -10 degrees Celsius. The durability of this system ensures that a system breakdown or replacement won't be a concern.

Longevity

Long battery life is a requirement for effective use of this solution. Therefore, the battery life of this prototype was carefully considered and extended. The transceiver uses only 2µA during sleep and 8mA when running for a maximum of 30 seconds per hour. The sensor is rated to use 15mA while on and is only powered for 30 seconds per hour. At this power level, the 10,000mAh battery can run the sensor and transceiver for almost 6 years. This could be further extended by reducing the frequency of sensor updates or by increasing the size of the battery. The long battery life means that it is not necessary to remove the sensors until it is time for harvest. These sensors can easily last a full growing season, even if additional sensors are added to get different data.

Ease of Use

This system is easy for farmers to implement into their workflow. Once installed, the system will need no further input from the farmer to function. Additionally, the farmer can monitor the current moisture level of each section of the farm easily by connecting to the base station. Since the system automatically turns on and off sprinklers using the relay connectors, it can be left on its own and does not need further maintenance once the thresholds are set. Setting up more sensors is extremely easy and only requires one value to be changed. This system does not require WiFi to be used and can be completely isolated from any outside connection and still work. This is extremely beneficial because many farmers cannot afford to have a WiFi network across their fields. Additionally, this means that it is also less subject to interference since in certain areas WiFi networks are extremely densely populated, allowing farmers to place these sensors wherever necessary, without regard to nearby technology that may affect the effectiveness of this sensor system.

This system is also very expandable. While a soil moisture sensor was used for this demonstration, any type of sensor can be used with this technology and with very minor to no tweaks.

Cost-Benefit Analysis

Table 2: Cost Analysis of the Sensor

Item	Name	Quantity	Cost per Unit	Total Cost
Transceiver	Moteino with RFM69HCW	1	$21.90	$21.90
Jumper Wires	ELEGOO Solderless Flexible Breadboard Jumper Wires 4 Different Lengths Male to Male	9	$0.09	$0.81
Connector Boards	GFORTUN 5cm x 7cm Double Sided 2.54mm Protoboard Prototyping PCB DIY Soldering Universal Printed Circuit Board	1	$0.78	$0.78

Moisture Sensor	High Sensitivity Soil Moisture Sensor Module with Female to Female Jump Wires	1	$1.50	$1.50
Battery	Portable Charger 10000mAh Power Bank High Capacity Power Bank	1	$7.00	$7.00
USB Cable	1FT Micro USB Cable	1	$1.40	$1.40
Total Cost	$33.38			

Table 3: Cost Analysis of the Base Station

Item	Name	Quantity	Cost per Unit	Total Cost
Transceiver	Moteino with RFM69HCW	1	$21.90	$21.90
Jumper Wires	ELEGOO Solderless Flexible Breadboard Jumper Wires 4 Different Lengths Male to Male	9	$0.09	$0.81
Connector Boards	GFORTUN 5cm x 7cm Double Sided 2.54mm Protoboard Prototyping PCB DIY Soldering Universal Printed Circuit Board	1	$0.78	$0.78
Relay	SunFounder 4 Channel 5V Relay Shield Module	1	$7.99	$7.99
LED	Clear LED Light Emitting Diodes Bulb LED Lamp, 5 mm	1	$0.08	$0.08
USB Cable	1FT Micro USB Cable	1	$1.20	$1.20
Total Cost	$32.75			

An average farm, according to the USDA, is 445 acres, this means that approximately 4 base stations are needed to cover the field. Additionally, other sensor manufacturers recommend 1 sensor every 40 acres, which translates to 11 sensors on this field. The total cost for this average farm would be $515. The Irrometer does not require base stations but would cost a total of $8250, and the Davis system would require 7 base stations to communicate with all the

sensors. The final cost of the Davis system would be over $10,000 without even including the actual sensors. In comparison, this prototype would cost..and therefore be significantly cheaper than currently available systems. Figure 9 shows a visual representation of the abilities of each system.

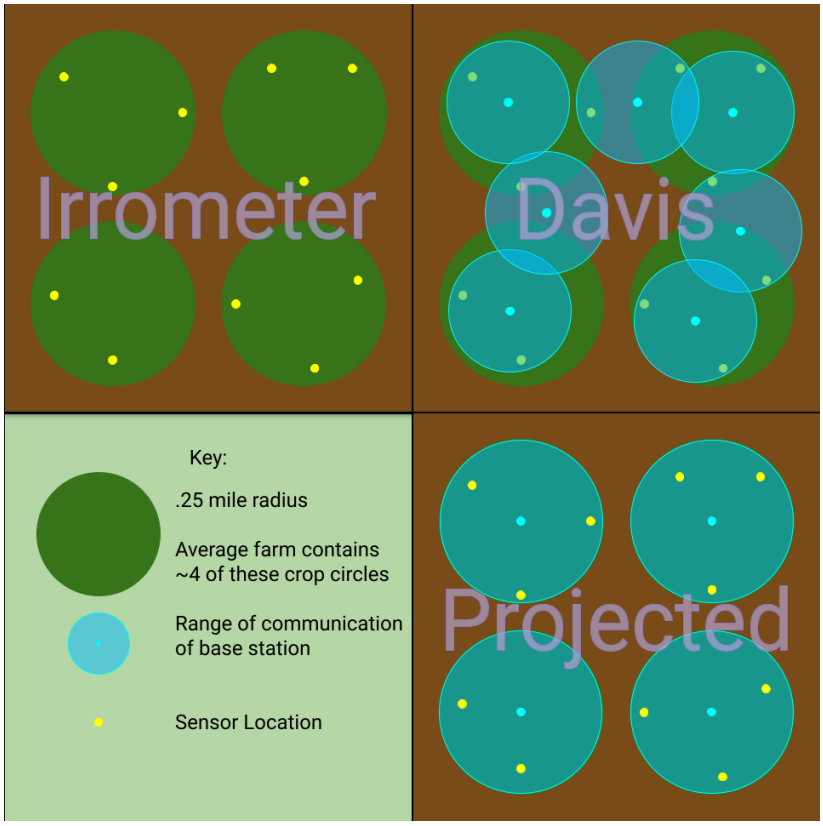

Figure 9: Diagram explaining the different available systems

Conclusion

All engineering goals for this project were met. The sensor system successfully measured and communicated in-field sensor data to a central base station. It exceeded the goal range of 100 meters and was able to communicate over 400 meters. The battery life of the sensor should last for years and will last for six months during a growing season. It was able to automatically control a relay to control the sprinkler system.

Significance

These results are significant because wireless transmission does not require the installation of wires, which can be costly and time-consuming. Additionally, its automatic

control of sprinklers means that the system can run without human input. Since it's cost-effective at only $83 for a sensor and base station, it will allow for a larger market of farmers to purchase this solution. Farms will be able to save some of the 3 trillion gallons of water wasted. This is especially important in drought-plagued areas such as California where every drop of water counts. Finally, with the reduction in water consumption, water rights allow excess water to be sold back to the water department for an extra income source.

Limitations

This device is only a prototype, so it has its limitations. Basic knowledge of Arduino is needed in its current form to change the transceiver IDs and to implement a different sensor. Since the battery life was not tested in the real world, it may be shorter than the calculated life. This may happen more if the area is exceptionally cold because lithium ion batteries do not work well in the cold.

Future Studies

A web-based application could be created to allow for more easily accessible data. The app could also remove the need for knowledge of Arduino since you can configure the id of the radio on the app. Solar panels could be implemented. Solar panels would allow for the battery to gain power during the day. It should allow for unlimited battery life if enough sun is present. Supercapacitors could be paired with solar panels such that there is no battery degradation. This would allow for a truly infinite battery life given enough sun. A larger antenna could also be used. This would allow for an even greater range and fewer base stations would be required.

Appendix

Appendix A: Base Station Code

```
#include <RFM69.h>    //get it here: https://www.github.com/lowpowerlab/rfm69
#include <SPI.h>
#include <SPIFlash.h> //get it here: https://www.github.com/lowpowerlab/spiflash

#define NODEID        1    //unique for each node on same network
#define NETWORKID     100  //the same on all nodes that talk to each other
```

```
        //Match frequency to the hardware version of the radio on your Moteino (uncomment one):
        // #define FREQUENCY     RF69_433MHZ
        //#define FREQUENCY     RF69_868MHZ
        #define FREQUENCY     RF69_915MHZ
        #define ENCRYPTKEY    "sampleEncryptKey" //exactly the same 16 characters/bytes on all nodes!
        #define IS_RFM69HW    //uncomment only for RFM69HW! Leave out if you have RFM69W!
        #define ACK_TIME      30 // max # of ms to wait for an ack
        #define SERIAL_BAUD   9600

        #ifdef __AVR_ATmega1284P__
          #define LED           15 // Moteino MEGAs have LEDs on D15
          #define FLASH_SS      23 // and FLASH SS on D23
        #else
          #define LED           9 // Moteinos have LEDs on D9
          #define FLASH_SS      8 // and FLASH SS on D8
        #endif

        #define RELAY 4
        RFM69 radio;
        SPIFlash flash(FLASH_SS, 0xEF30); //EF30 for 4mbit  Windbond chip (W25X40CL)
        bool promiscuousMode = false; //set to 'true' to sniff all packets on the same network

        void setup() {
          Serial.begin(SERIAL_BAUD);
          delay(10);
          radio.initialize(FREQUENCY,NODEID,NETWORKID);
        #ifdef IS_RFM69HW
          radio.setHighPower(); //only for RFM69HW!
        #endif
          radio.encrypt(ENCRYPTKEY);
          // radio.promiscuous(promiscuousMode);
          //radio.setFrequency(919000000);
          char buff[50];
              sprintf(buff, "\nListening at %d Mhz...", FREQUENCY==RF69_433MHZ ? 433 :
FREQUENCY==RF69_868MHZ ? 868 : 915);
          Serial.println(buff);
```

```
  if (flash.initialize())
  {
    Serial.print("SPI Flash Init OK. Unique MAC = [");
    flash.readUniqueId();
    for (byte i=0;i<8;i++)
    {
      Serial.print(flash.UNIQUEID[i], HEX);
      if (i!=8) Serial.print(':');
    }
    Serial.println(']');

    //alternative way to read it:
    //byte* MAC = flash.readUniqueId();
    //for (byte i=0;i<8;i++)
    //{
    //  Serial.print(MAC[i], HEX);
    //  Serial.print(' ');
    //}

  }
  else
    Serial.println("SPI Flash Init FAIL! (is chip present?)");
}

byte ackCount=0;
uint32_t packetCount = 0;
void loop() {
  // Serial.println(analogRead(A0));
  //process any serial input
  if (Serial.available() > 0)
  {
    char input = Serial.read();
    if (input == 'r') //d=dump all register values
      radio.readAllRegs();
    if (input == 'E') //E=enable encryption
      radio.encrypt(ENCRYPTKEY);
```

```
if (input == 'e') //e=disable encryption
  radio.encrypt(null);
if (input == 'p')
{
  promiscuousMode = !promiscuousMode;
  // radio.promiscuous(promiscuousMode);
  Serial.print("Promiscuous mode ");Serial.println(promiscuousMode ? "on" : "off");
}

if (input == 'd') //d=dump flash area
{
  Serial.println("Flash content:");
  int counter = 0;

  while(counter<=256){
    Serial.print(flash.readByte(counter++), HEX);
    Serial.print('.');
  }
  while(flash.busy());
  Serial.println();
}
if (input == 'D')
{
  Serial.print("Deleting Flash chip ... ");
  flash.chipErase();
  while(flash.busy());
  Serial.println("DONE");
}
if (input == 'i')
{
  Serial.print("DeviceID: ");
  word jedecid = flash.readDeviceId();
  Serial.println(jedecid, HEX);
}
if (input == 't')
{
```

```
            byte temperature =  radio.readTemperature(-1); // -1 = user cal factor, adjust for correct ambient
            byte fTemp = 1.8 * temperature + 32; // 9/5=1.8
            Serial.print( "Radio Temp is ");
            Serial.print(temperature);
            Serial.print("C, ");
             Serial.print(fTemp); //converting to F loses some resolution, obvious when C is on edge between 2 values (ie 26C=78F, 27C=80F)
            Serial.println('F');
          }
        }

        if (radio.receiveDone())
        {
          Serial.print("#[");
          Serial.print(++packetCount);
          Serial.print(']');
          Serial.print('[');Serial.print(radio.SENDERID, DEC);Serial.print("] ");
          if (promiscuousMode)
          {
            Serial.print("to [");Serial.print(radio.TARGETID, DEC);Serial.print("] ");
          }
          // Serial.print(radio.DATA);
          for (byte i = 0; i < radio.DATALEN; i++)
            Serial.print(radio.DATA[i]);
            // Serial.print(radio.DATALEN);
          Serial.print("   [RX_RSSI:");Serial.print(radio.RSSI);Serial.print("]");

          if (radio.ACKRequested())
          {
            byte theNodeID = radio.SENDERID;
            radio.sendACK();
            Serial.print(" - ACK sent.");

            // When a node requests an ACK, respond to the ACK
            // and also send a packet requesting an ACK (every 3rd one only)
            // This way both TX/RX NODE functions are tested on 1 end at the GATEWAY
```

```
      if (ackCount++%3==0)
      {
        Serial.print(" Pinging node ");
        Serial.print(theNodeID);
        Serial.print(" - ACK...");
        delay(3); //need this when sending right after reception
        if (radio.sendWithRetry(theNodeID, "ACK TEST", 8, 0))  // 0 = only 1 attempt, no retries
          Serial.print("ok!");
        else Serial.print("nothing");
      }
    }
    Blink(LED,5);
    Serial.print(radio.DATA[2]);
    if(radio.DATA[2]<210){
      pinMode(LED, OUTPUT);
      digitalWrite(LED,HIGH);
      pinMode(RELAY, OUTPUT);
      digitalWrite(RELAY,HIGH);
      Serial.print("1");
    }else{
      pinMode(LED, OUTPUT);
      digitalWrite(LED,LOW);
      pinMode(RELAY, OUTPUT);
      digitalWrite(RELAY,LOW);
      Serial.print("0");
    }
    Serial.println();
  }
}

void Blink(byte PIN, int DELAY_MS)
{
  pinMode(PIN, OUTPUT);
  digitalWrite(PIN,HIGH);
  delay(DELAY_MS);
  digitalWrite(PIN,LOW);
```

```
}
```

Appendix B: Sensor Station Code

```
#include <RFM69.h>        //get it here: https://www.github.com/lowpowerlab/rfm69
#include <RFM69_ATC.h>    //get it here: https://www.github.com/lowpowerlab/rfm69
#include <SPIFlash.h>     //get it here: https://www.github.com/lowpowerlab/spiflash
#include <SPI.h>          //included with Arduino IDE install (www.arduino.cc)
#include <LowPower.h>
#define NODEID      99 // Ensure this is unique
#define NETWORKID   100 // Ensure this is same on all
#define GATEWAYID   1 // Ensure this matches with the gateway
//Match frequency to the hardware version of the radio on your Moteino (uncomment one):
//#define FREQUENCY     RF69_433MHZ
//#define FREQUENCY     RF69_868MHZ
#define FREQUENCY     RF69_915MHZ
#define ENCRYPTKEY    "sampleEncryptKey" //has to be same 16 characters/bytes on all nodes, not more not less!
#define IS_RFM69HW_HCW  //uncomment only for RFM69HW/HCW! Leave out if you have RFM69W/CW!

#define SERIAL_BAUD 9600

#ifdef ENABLE_ATC
  RFM69_ATC radio;
#else
  RFM69 radio;
#endif

#ifdef __AVR_ATmega1284P__
  #define LED           15 // Moteino MEGAs have LEDs on D15
  #define FLASH_SS      23 // and FLASH SS on D23
#else
  #define LED           9 // Moteinos have LEDs on D9
  #define FLASH_SS      8 // and FLASH SS on D8
```

```
#endif

SPIFlash flash(FLASH_SS, 0xEF30); //EF40 for 16mbit windbond chip


int TRANSMITPERIOD = 250; //transmit a packet to gateway 4 times every second
int MAXATTEMPTS = 30000/TRANSMITPERIOD; //attempt for 30 seconds
int attempts = 0;
byte sendSize=0;

typedef struct {
  int          nodeId; //store this nodeId
  int sensor; //store the sensor data
} Payload;
Payload theData;


void setup() {
  Serial.begin(SERIAL_BAUD);
  radio.initialize(FREQUENCY,NODEID,NETWORKID);
#ifdef IS_RFM69HW_HCW
  radio.setHighPower(); //must include this only for RFM69HW/HCW!
#endif
  radio.encrypt(ENCRYPTKEY);
  char buff[50];
      sprintf(buff, "\nTransmitting at %d Mhz...", FREQUENCY==RF69_433MHZ ? 433 :
FREQUENCY==RF69_868MHZ ? 868 : 915);
  Serial.println(buff);

  if (flash.initialize())
    Serial.println("SPI Flash Init OK!");
  else
    Serial.println("SPI Flash Init FAIL! (is chip present?)");
}

long lastPeriod = -1;
```

```
void loop() {

  int currPeriod = millis()/TRANSMITPERIOD;
  if (currPeriod != lastPeriod)
  {
    //fill in the struct with new values
    theData.nodeId = NODEID;
    theData.sensor = analogRead(A0);

    lastPeriod=currPeriod;
    delay(100);
        if (radio.sendWithRetry(GATEWAYID, (const void*)(&theData), sizeof(theData)) ||
attempts>MAXATTEMPTS){
      //Sleep for 1 hour
      LowPower.longPowerDown(3600000);
      //Reset the number of attempts
      attempts = 0;
    }
    else attempts++;
  }
}
```

Bibliography

Ahmed, M. A., Gallardo, J. L., Zuniga, M. D., Pedraza, M. A., Carvajal, G., Jara, N., & Carvajal, R. (2022). LoRa Based IoT Platform for Remote Monitoring of Large-Scale Agriculture Farms in Chile. *Sensors*, *22*(8), 2824. https://doi.org/10.3390/s22082824

Augustin, A., Yi, J., Clausen, T., & Townsley, W. (2016). A Study of LoRa: Long Range & Low Power Networks for the Internet of Things. *Sensors*, *16*(9), 1466. https://doi.org/10.3390/s16091466

Cattani, M., Boano, C., & Römer, K. (2017). An Experimental Evaluation of the Reliability of LoRa Long-Range Low-Power Wireless Communication. *Journal of Sensor and Actuator Networks*, *6*(2), 7. https://doi.org/10.3390/jsan6020007

Chen, C. W., & Wang, Y. (2008). Chain-Type Wireless Sensor Network for Monitoring Long Range Infrastructures: Architecture and Protocols. *International Journal of Distributed Sensor Networks*, *4*(4), 287–314. https://doi.org/10.1080/15501320701260261

Choi, K. W., Ginting, L., Aziz, A. A., Setiawan, D., Park, J. H., Hwang, S. I., . . . Kim, D. I. (2019). Toward Realization of Long-Range Wireless-Powered Sensor Networks. *IEEE Wireless Communications*, *26*(4), 184–192. https://doi.org/10.1109/mwc.2019.1800475

Fisher, D., Woodruff, L., Anapalli, S., & Pinnamaneni, S. (2018). Open-Source Wireless Cloud-Connected Agricultural Sensor Network. *Journal of Sensor and Actuator Networks*, *7*(4), 47. https://doi.org/10.3390/jsan7040047

G. Evans, R. (2001). *Center Pivot Irrigation*. United States Department of Agriculture. https://www.ars.usda.gov/ARSUserFiles/21563/center%20pivot%20design%202.pdf

Georgiou, O., & Raza, U. (2017). Low Power Wide Area Network Analysis: Can LoRa Scale? *IEEE Wireless Communications Letters*, *6*(2), 162–165. https://doi.org/10.1109/lwc.2016.2647247

Grassini, P., Yang, H., Irmak, S., Thorburn, J., Burr, C., & Cassman, K. G. (2011). High-yield irrigated maize in the Western U.S. Corn Belt: II. Irrigation management and crop water productivity. *Field Crops Research*, *120*(1), 133–141. https://doi.org/10.1016/j.fcr.2010.09.013

John, J., Kasbekar, G., Sharma, D., Ramulu, V., & Baghini, M. (2018). Design and Implementation of a Wireless Sensor Network for Agricultural Applications. *EAI*

Endorsed Transactions on Internet of Things, *4*(16), 158420. https://doi.org/10.4108/eai.13-7-2018.158420

Malik, N. N., Alosaimi, W., Uddin, M. I., Alouffi, B., & Alyami, H. (2020). Wireless Sensor Network Applications in Healthcare and Precision Agriculture. *Journal of Healthcare Engineering*, *2020*, 1–9. https://doi.org/10.1155/2020/8836613

Maraveas, C., & Bartzanas, T. (2021). Sensors for Structural Health Monitoring of Agricultural Structures. *Sensors*, *21*(1), 314. https://doi.org/10.3390/s21010314

Matin, M., & Islam, M. (2012). Overview of Wireless Sensor Network. *Wireless Sensor Networks - Technology and Protocols*. https://doi.org/10.5772/49376

Morais, R., Mendes, J., Silva, R., Silva, N., Sousa, J. J., & Peres, E. (2021). A Versatile, Low-Power and Low-Cost IoT Device for Field Data Gathering in Precision Agriculture Practices. *Agriculture*, *11*(7), 619. https://doi.org/10.3390/agriculture11070619

Navulur, S., S. C. S. Sastry, A., & N. Giri Prasad, M. (2017). Agricultural Management through Wireless Sensors and Internet of Things. *International Journal of Electrical and Computer Engineering (IJECE)*, *7*(6), 3492. https://doi.org/10.11591/ijece.v7i6.pp3492-3499

San-Um, W., Lekbunyasin, P., Kodyoo, M., Wongsuwan, W., Makfak, J., & Kerdsri, J. (2017). A long-range low-power wireless sensor network based on U-LoRa technology for tactical troops tracking systems. *2017 Third Asian Conference on Defence Technology (ACDT)*. https://doi.org/10.1109/acdt.2017.7886152

Shafiq, M., Ahmad, M., Irshad, A., Gohar, M., Usman, M., Khalil Afzal, M., . . . Yu, H. (2018). Multiple Access Control for Cognitive Radio-Based IEEE 802.11ah Networks. *Sensors*, *18*(7), 2043. https://doi.org/10.3390/s18072043

Šljivo, A., Kerkhove, D., Tian, L., Famaey, J., Munteanu, A., Moerman, I., . . . De Poorter, E. (2018). Performance Evaluation of IEEE 802.11ah Networks With High-Throughput Bidirectional Traffic. *Sensors*, *18*(2), 325. https://doi.org/10.3390/s18020325

Sun, W., Choi, M., & Choi, S. (2017). IEEE 802.11ah: A Long Range 802.11 WLAN at Sub 1 GHz. *Journal of ICT Standardization*, *1*(1), 83–108. https://doi.org/10.13052/jicts2245-800x.115

Swain, M., Zimon, D., Singh, R., Hashmi, M. F., Rashid, M., & Hakak, S. (2021). LoRa-LBO: An Experimental Analysis of LoRa Link Budget Optimization in Custom Build IoT Test

Bed for Agriculture 4.0. *Agronomy*, *11*(5), 820. https://doi.org/10.3390/agronomy11050820

USDA ERS - Irrigation & Water Use. (n.d.). Retrieved from https://www.ers.usda.gov/topics/farm-practices-management/irrigation-water-use/

USDA, National Agricultural Statistics Service. (2018). *Census of Agriculture* [Dataset]. Retrieved from https://www.nass.usda.gov/Publications/AgCensus/2017/Online_Resources/Farm_and_Ranch_Irrigation_Survey/fris_1_0023_0023.pdf

Wichelns, D. (2010). Agricultural Water Pricing: United States. *Organization for Economic Co-Operation and Development*. Retrieved from https://www.oecd.org/unitedstates/45016437.pdf

Yanru Wang, Kok Keong Chai, Yue Chen, & John Schormans. (2017). Energy Efficient Window Control Scheme for IEEE 802.11ah (Wi-Fi HaLow) Based Networks. *J. Of Electrical Engineering*, *5*(5). https://doi.org/10.17265/2328-2223/2017.05.003