Vol. 7(2), pp. 17-25, July-December, 2023
DOI: 10.5897/JECI2022.0205
Article Number: A973F1271109
ISSN: 2141-6508

http://www.academicjournals.org/JECI

ACADEMIC JOURNALS
expand your knowledge

Journal of Engineering and Computer Innovations

Full Length Research Paper

LoRa flood messaging applied to remote soil-moisture monitoring

Peter Raeth[1]* and Philip Branch[2]

[1]Creative Solutions, Senior Research Engineer, Beavercreek Ohio USA.
[2]Faculty of Science, Engineering and Technology, Swinburne University of Technology, Melbourne 3122, VIC, Australia.

Received 7 November, 2022; Accepted 30 July, 2023

Soil-Moisture sensing has proven to be an important factor in decreasing the need for irrigation while maintaining crop yield and quality. Farmers' finding ways to minimize water use is necessary given the world's obvious climate change and, in some areas, the resulting onslaught of drought. As the world's population grows and the demand for higher-quality food increases, precision irrigation becomes even more essential. As sensors are deployed, these have to be monitored in some way. Our project employs radio broadcasts over unlicensed frequencies to implement wireless remote sensing. Internet and cellular are alternative solutions. However, those require more than the acquisition and implementation of certain equipment and software. Those two approaches also involve third-party services, and their continuing fees. An additional assumption is that those services are available, cost effective, and reliable. Radio broadcast using the LoRa (Long Range) protocol is essentially a low-power means for implementing transceivers for low-rate data exchange. These are effective over longer ranges than technologies such as Bluetooth. LoRa is often deployed using LoRaWAN (LoRa for formal Wide Area Networks) but that involves additional equipment. This paper discusses a means of using only relatively low-cost hardware (Arduino-Uno microcontrollers with attached Dragino transceivers). A wireless network is formed upon which messages flow from sensors attached to microcontrollers, through relays (to account for terrain variability and other obstructions), to a receiver connected to some external system. The LoRa hardware unit remains consistent across all components, with only the installed software varying. Relays and receivers utilize their own unchanging software. However, the sensor units' baseline software is adapted based on the types and numbers of sensors attached. Each sensor, relay, and receiver has its own unique assigned network address. The authors have constructed and demonstrated a proof-of-concept of the ideas expressed in this paper. Discussion is offered on future work required as the project moves toward field application.

Key words: Precision agriculture, precision irrigation, LoRa, IoT, conservation, soil moisture, remote sensing, flood messaging, drought, wireless network.

INTRODUCTION

Precision irrigation is a component of precision agriculture. Agricultural knowledge is combined with other spheres such as computer engineering, geographic information systems, remote sensing, electronics, and

*Corresponding author. E-mail: peter_raeth@juno.com.

meteorology. The goal is to improve, or at least maintain, crop value while using fewer resources such as land, pesticides, water, and fertilizers. It does so while retaining nutrition and without resorting to the risks of genetic mutation.

Given the considerable literature in precision agriculture (Oliver et al., 2013; Lal and Steward, 2016; Zhang, 2016; Shannon et al., 2018), it is clear that combining other fields of work with agriculture can help improve agrarian results.

Precision agriculture's positive impact has been noted by Atalla et al. (2023), Ncube et al. (2018), as well as Jacobs et al. (2018). Within precision irrigation, sensing and control mechanisms have the potential to aid countries stricken by drought as those countries move beyond their dependence on direct-rain for watering crops. This could extend opportunities to grow crops outside the rainy season, while still not over-farming the land.

Besides growing seasons limited by direct-rain irrigation, drought requires careful use of water. This leads naturally to control over irrigation so that crops are not over-watered or under-watered. Precision irrigation seeks to improve over methods that rely on fixed schedules and fixed volumes. Based on plant, soil, terrain, and ambient weather conditions, measures are taken to estimate a crop's water needs and irrigate on that basis. In the same way that crop health cannot always be estimated by visual observation, the same is true of a crop's water need. By the time the crop or land shows visual signs of distress, the time when irrigation should have been applied is past.

An approach one often sees in the literature is soil-moisture sensing as the sole basis for deciding when and by how much to irrigate. Of particular interest in this domain is the work of Munyaradzi and his team (2013, 2022, 2023 TBP). For years they have successfully operated a practical system of this type on an open-air farm under uncontrolled conditions. Other indicators of this approach's success are evident (Or, 2005; Kisekka et al., 2014; Svedberg, 2019; Millán et al., 2019; Zafar, Arshad et al., 2020).

Having established the value of precision agriculture and precision irrigation, one is led to the issues of sensors and establishing access to those sensors. Internet and cellular telecon are two oft-seen approaches for sending sensor data to their point of use. These assume the existence of such services and that those services are of sufficient speed, reliability, and cost effectiveness. Such services are operated by third-parties and one has to pay continuing fees to use them. There are numerous successful examples of precision irrigation using internet and cellular telecon. Petousi et al. (2018) employ the internet to link system components, as did Chin and Audah (2017). Munyaradzi et al. (2022) use cellular telecon.

The authors' approach uses unlicensed radio frequencies and employs the LoRa protocol (Montagny, 2022). To avoid the expense and complexity of LoRaWAN, a flood-messaging system is carried out by relatively low-cost Arduino /Dragino microcontroller/ transceiver nodes. The very same node hardware is employed throughout, with differences in software, depending on the purpose of a specific node and the sensors connected to it, if any. The resulting network operates stand-alone. The receiver and basestation can be hosted by a stand-alone computer. Still, various components could be hosted at different locations on a local-area network, the internet, or a cellular telecon system. This flexibility allows many implementation options.

Related work

Bluetooth is another wireless protocol that uses unlicensed radio frequencies and operates over shorter ranges than LoRa. Kim and Evans (2009) used this approach to create a self-contained system of sensors and basestations. Access to internet and cellular telecon is not required by their system but, as does this present work, allows for third-party services as those are needed. Their earlier work (Kim et al., 2008) offers considerable detail on their successful implementation.

Where this present project evolves a routeless flood-messaging network, Nootropic Design (2018) employed a mesh network, within which routes can be discovered and established. That and the present project employ LoRa and avoid LoRaWAN. Both use relatively inexpensive microcontroller/transceiver nodes to create self-healing networks. Our brief foray into mesh networks showed that the Arduino-Uno's memory is almost totally occupied by the mesh libraries. A higher-cost version beyond Uno (Mega) would be necessary.

MATERIALS AND METHODS

Equipment and tools

Developing and testing the network described in this paper used the following equipment and tools:

1) Arduino Uno R3 microcontroller board and its associated USB cable. Four were used. It is possible that the Arduino Uno R4 or the Arduino Mega could be used in direct replacement.
2) Dragino LoRa Transceiver. Four were used. These plug directly into a microcontroller board.
3) A Windows or Linux PC. This acts as a basestation. One was used.
4) PyCharm Community. This is a free open-access tool for working with Python code.
5) Arduino IDE. This is a free open-access tool for working with Arduino C++ code.

System implementation

The authors began with the flood-messaging system designed,

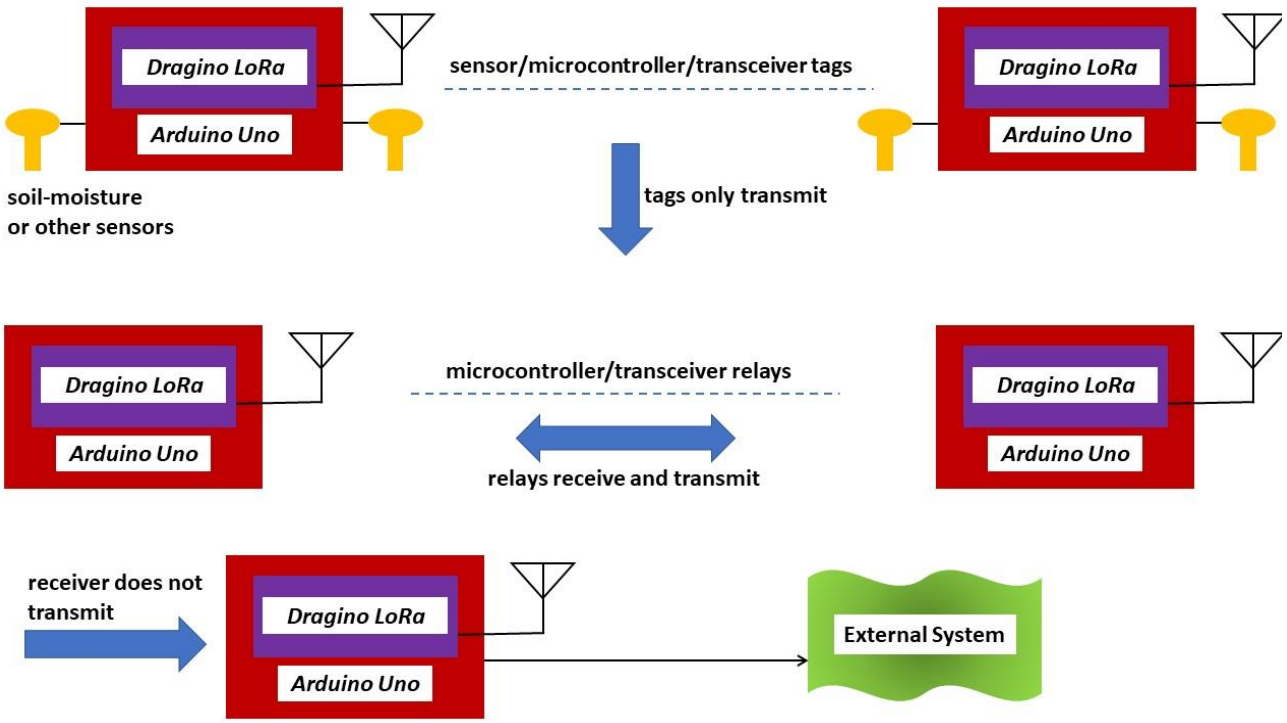


Figure 1. Top-Level network layout.
Source: Author

implemented, and tested by Branch et al. (2020). They provide a detailed explanation of the theory behind their messaging system. Their work resulted in a LoRa-based, sensor network for transmitting location information of personnel and equipment in an underground facility. In this present work, their network is applied to remote soil-moisture monitoring in such a way that their approach can be applied in the general case of remote-sensing (Sensors produce a voltage which is then converted to a digital value. The digital value is sent via some messaging system, monitored, and interpreted relative to the sensor involved. The interpretation of data leads to some action relative to the situation at hand).

The basic layout of the network is shown in Figure 1. There are three types of nodes in the network: tag, relay, and receiver. Hardware for all nodes is always the same, Arduino-Uno microcontroller, into which is plugged a Dragino LoRa transceiver. Both are readily-available and low-cost (about $75US per node, delivered). This is the only hardware required to establish the network. The expense and complication of LoRaWAN are avoided, as are third-party services.

Tag nodes transmit data delivered by soil-moisture sensors. Relay nodes receive and retransmit messages. The single receiver node intakes messages and operates according to the data extracted. It also connects to an external system. That connection could be via Ethernet, Internet, cellular telecon, USB, RS232, or other means. There is a lot of flexibility built into the receiver such that no additional network is required. The receiver could be plugged into a stand-alone PC. The PC could be running either Windows or Linux. MacOS is another possibility but the authors have not tested that option. Testing was with Windows-10 and Ubuntu-Linux. Additional detail on Tags, Relays, and the Receiver is given in the following subsections.

All nodes are constructed from Arduino/Dragino pairs. The Dragino module plugs into the Arduino module. The pair is illustrated in Figure 2.

Tags

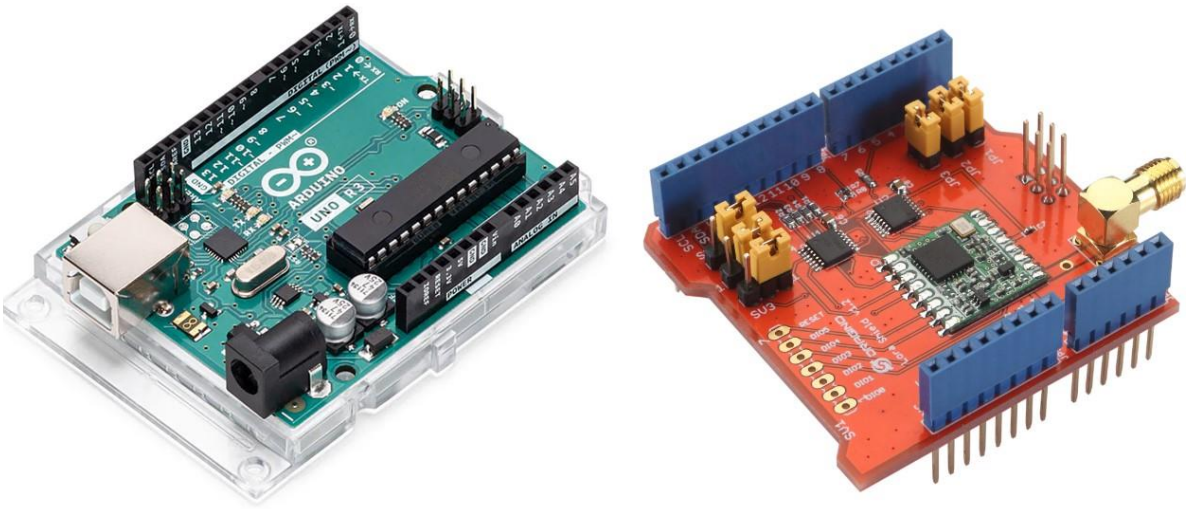
Tags employ the basic algorithm reflected in Figure 3. “Message information” includes a unique message ID (sequence). While the algorithm is apparently simple, there has to be unique code to carry out “read sensor value”. At their project link, the authors demonstrate code for a particular soil-moisture sensor. Other sensors will require different code. The details behind connecting and programming for the sensors used in our testing are given in the documentation associated this project. See Raeth (2023) for details on circuitry associated with connecting a soil-moisture sensor. "Be aware of the need to calibrate such sensors, as much depends on the specific soil characteristics and the type of sensor used. Calibration procedures that take these factors into account are provided by Hrisko (2020) and Edaphic (2023). In this paper, the authors do not focus on soil-moisture sensing itself but rather on the data transfer methods to move the resulting data from one place to another."

Relays

Software for all relays is always the same. The underlying algorithm is exposed in Figure 4. All messages received by a relay are retransmitted, as long as the message’s time-to-live (TTL) is greater than zero. In this way, messages are “forwarded” within the network.

Receiver

The receiver is a unique node within the network and Figure 5 reflects its algorithm. It does not transmit at all. New messages are



• Arduino Uno R3 Microcontroller Board (left)
Source: https://store-usa.arduino.cc/products/arduino-uno-rev3

* Dragino LoRa Transceiver (right)
Source: https://www.dragino.com/products/lora/item/102-lora-shield.html

Figure 2. Arduino/Dragino pair.

```
while true do
    for each sensor associated with this tag
        wait(a few seconds)
        read sensor value
        increment sensor's message sequence
        assign message information
        assign message data
        transmit(message)
```

Figure 3. Pseudo-code representation of logic employed by tag.
Source: Author

sent to the external system. One will notice that, in many ways, the receiver is much like a relay, except that it does not forward messages.

External system

Our receiver plugs into a PC's USB port. Once software on the PC connects to the serial port opened by the Arduino-Uno microcontroller, the microcontroller can send messages through that port to the external system. The external system can do anything it likes with those messages and the data elements (fields) contained therein.

RESULTS AND DISCUSSION

All documentation and software for this project is available at LoRaFloodMessaging within https://github.com/SoothingMist/Scalable-Point-to-Point-LoRa-Sensor-Network.

Demonstration

In their earlier work, Branch et al. (2020) provided an exposition and a statistical analysis of messages flowing in their flood-messaging network. This section exposes certain operational aspects in the present version of the network. Tag nodes generate and broadcast fixed-length, space-separated, messages having these elements:

1) PREFIX: A standard prepend to identify the network that spawned the message. As there is only with one network, this value is a constant.
2) SEQUENCE: An incrementing value so that a history of messages can be established.
3) TYPE: Always equal to 0 since this is an original tag-generated. Type 9 is a reset which is sent upon tag startup.
4) TAGID: A network-unique ID assigned to each sensor.
5) RELAYID: Always equals 0 for tag-generated messages. (Relay IDs start at 1.)
6) TTL: Time-To-Live always initializes at 5 for our purposes and indicates the number of acceptable forwardings by relays.
7) RECEIVED_RSSI: Received Signal Strength Indication

```
while true do
    while channel is idle
        monitor channel for message
    upon receipt of message
        if message.sequence <= tag[message.tag].sequence
            discard message
        else
            if message.type eq Reset
                tag[message.tag].sequence = 0
            else
                tag[message.tag].sequence = message.sequence
            message.ttl = message.ttl – 1
            if message.ttl eq 0
                discard message
            else
                update message data fields as needed
                wait(exponentially distributed random amount of time, mean 100 ms)
                transmit(message)
```

Figure 4. Pseudo-code representation of logic employed by relay.
Source: Author

```
while true do
    while channel is idle
        monitor channel for message
    upon receipt of message
        if message.sequence <= tag[message.tag].sequence
            discard message
        else
            if message.type eq Reset
                tag[message.tag].sequence = 0
            else
                tag[message.tag].sequence = message.sequence
                send message to external system
```

Figure 5. Pseudo-code representation of logic employed by receiver.
Source: Author

always equals 0 when a node originates a message.
8) INFO: Gives information about the sensor data. For instance, one could put the sensor type and microcontroller pin connection here.
9) Sensordatastr: The sensor data itself.

Relay nodes rebroadcast all new messages received, decrementing TTL prior to each rebroadcast. Once TTL equals 0, the message dies. Old messages are identified by a tag's lower sequence number. RSSI from the received packet is added to each rebroadcast message. The receiver node forwards all new messages to the external system.

This arrangement was tested using one receiver, one relay, and two tag nodes, each of which had three sensors associated with it. Connection to an external system that enables collection, analysis, presentation, and action is essential to deriving value from remote sensing. If there is no focused action, relative to goals, there is little value to be had from data gathering.

All software in our external system is written in Python. Web-based languages are also employed, although use of the internet is not at all necessary. It is just that a browser is used for display. An html address is just as valid on a stand-alone (not networked) PC as it is on a local-area network or a world-wide network such as the internet. Arduino-Uno is programmed in C/C++, although its version of that language is not always according to the standard. It is the same with other microcontrollers which use that language.

One simple external system is demonstrated using the PC's command console. A brief program was constructed to repeat all messages from the receiver node. An example is shown in Figure 6. The receiver is connected to the PC via USB.

Another version of the external system opens itself to network queries via the REST interface (TutorialsPoint, 2022). Once a query is received, the oldest message is forwarded to the requesting address. As this is assumed to be a single-user system, it responds only to a specific requesting address. An example of a query response is shown in Figure 7.

Any process issuing a REST request to the appropriate address receives a response similar to that shown. The process then parses the message and carries on from there.

Query responses can be used in numerous ways, as the situation demands. A base station is employed that can be hosted locally or remotely. Data from a selected sensor is plotted, along with a list of query responses. Figure 8 offers an example.

Future work

While not addressing networking issues, and using hard-wired sensor inputs, Guerbaoui et al. (2013) delve deeply into the mechanisms of automated irrigation. This present work only addresses the networking of sensors with a basestation.

The base station could be used to not only display sensor values but also to make decisions that would lead to actuation of an irrigation system to control volume and timing according the needs of a particular crop.

Network security and message integrity are another area of work. Cybersecurity is a vast domain that people spend their entire careers learning about and enabling. AES encryption/decryption was demonstrated but not incorporated.

That could easily be added to the present implementation. It was interesting to observe how much easier it was for encrypted messages to become corrupted as data flows through the network. This indicates a need for error detection/correction schemes.

Rudimentary message integrity was incorporated but that could be further addressed by adding constraints on message length and content. CRC codes would also help.

Time-To-Live (TTL) is a concern as the network grows and relay nodes are positioned to account for obstructions and distance. How long should a message live in the network? How many times should a message be rebroadcast? These are important questions to investigate.

A related issue is whether the receiver should be able to request a rebroadcast of a sensor's reading. That would require the receiver to originate messages and tags to take in messages. In a sense, tags would also act as receivers. Relays would not have to change. This adds considerable code complexity.

Power consumption of the various nodes is a concern. It is possible to operate the nodes for three years on one small 2000mah lithium battery charge. But, the mechanism used for that is not precise in its setting. For soil-moisture sensing, one might anticipate taking readings every six hours. The network would remain up only for as long as it takes for all original tag messages to arrive at the external system. Can the entire network be brought up and put to sleep at the same exact time? Certain microcontrollers have this capability but not Arduino-Uno. External circuitry would be needed.

Use of external rechargers such as solar or wind add expense to the system. Microcontroller memory utilization is also an issue. That has been managed well but, as the software and network grows; there are limits to what can be done. For now, it is important to not purchase more-expensive microcontroller boards. Making the software increasingly-difficult to understand in an attempt to save memory should only be reluctantly embraced. Advice provided by Arduino (2022) has proven to be quite valuable.

Consideration of system range is important along the path to field application. It is not possible to estimate a range for LoRa without considering operating frequency, type of antenna, spreading factor, electromagnetic interference, and physical obstructions. At this point in the authors' development process, we are concentrating

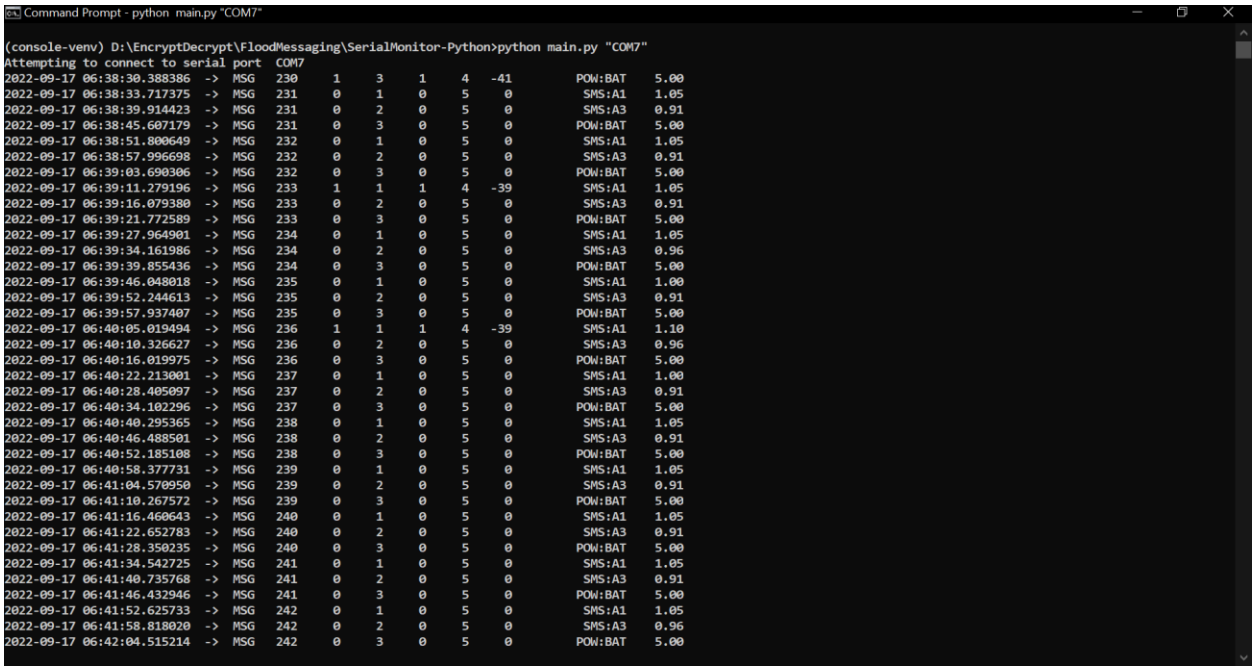


Figure 6. Echoing messages a PC gets from the receiver.
Source: Author

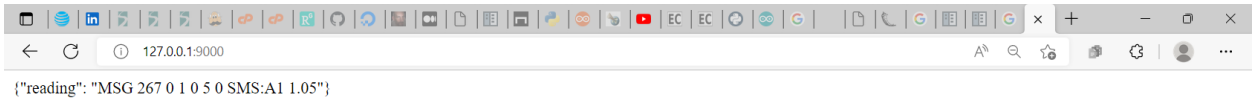


Figure 7. Example response to REST query.
Source: Author

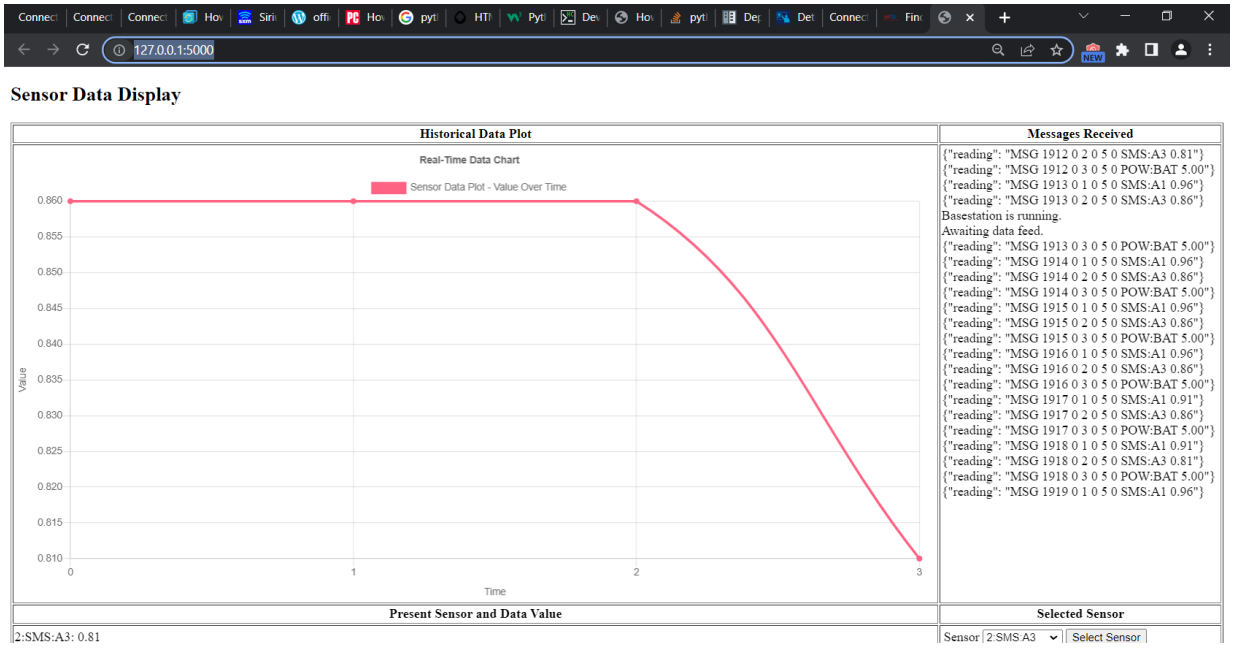


Figure 8**.** Basestation display of data and messages.
Source: Author

on network protocols and message content. We have not yet performed range tests within a given scenario. Yet, others have done so. Good discussions on transceiver range are provided in documentation provided by Arduino (2023), Yosensi (2021), and Semtech (2023). From those, it is clear that our particular equipment has sufficient range to satisfy our requirements, especially if relay nodes are employed. Edge Collective (2020) has indicated a direction that might be taken for our own testing.

The matter of weather-proofing cannot be ignored. Suitable casings for all electronics have to be identified and employed.

Conclusions

The authors implemented a flood-messaging system that enables an expanding self-healing network that accounts for distance and obstructions. This is accomplished without the expense and complexity of LoRaWAN or third-party services. Open-Access software and programming tools are used entirely. Arduino-Uno is open-access hardware, whereas Dragino is proprietary. There are other LoRa transceivers but Dragino has proven to be very reliable and compatible with Arduino-Uno due to the care of its manufacturing and documentation.

This project has shown that external systems can be hosted in a very flexible way, entirely on a stand-alone PC, distributed across a local-area network, or remotely across a world-wide network such as the internet.

The implementation of the network is very general in that it can easily accommodate sensors of various types. For instance, some practitioners use one soil-moisture sensor and one soil-temperature sensor per microcontroller. Others add weather data from a locally-installed station. The network does not restrict the type of sensor as long as the microcontroller is programmed to read and interpret the sensor at hand.

With simplicity and low expense it is possible to produce an effective network using radio broadcasts over unlicensed frequencies. The project has applied this network to soil-moisture monitoring.

ACKNOWLEDGEMENTS

The authors wish to thank Dr. Binghao Li and Dr. Kai Zhao from the Faculty of Engineering, University of New South Wales, Sydney, Australia, for their discussions. It also highlights the value of collaboration between scientists and engineers in transitioning theory to practical applications.

CONFLICT OF INTERESTS

The authors have not declared any conflict of interests.

REFERENCES

Arduino (2022). Reduce the size and memory usage of your sketch. Arduino Documentation. Available at: https://support.arduino.cc/hc/en-us/articles/360013825179

Arduino (2023). The Arduino Guide to LoRa and LoRaWAN. Arduino Documentation. Available at: https://docs.arduino.cc/learn/communication/lorawan-101

Atalla S, Tarapiah S, Gawanmeh A, Daradkeh M, Mukhtar H, Himeur Y, Mansoor W, Hashim KFB, Daadoo M (2023). IoT-Enabled precision agriculture: Developing an ecosystem for optimized crop management. Information 14(4). Available at: https://www.mdpi.com/2078-2489/14/4/205#:~:text=Precision%20agriculture%20is%20a%20modern,%2C%20and%20efficiency%20%5B45%5D

Branch P, Li B, Zhao K (2020). A LoRa-based linear sensor network for location data in underground mining. MDPI, Telecom 1(2):68-79.

Chin YS, Audah L (2017). Vertical farming monitoring system using the internet of things. AIP Conference Proceedings. Available at: https://aip.scitation.org/doi/pdf/10.1063/1.5002039

Edaphic (2023). Soil moisture sensor calibration. Documentation. Available at: https://edaphic.com.au/soil-water-compendium/soil-moisture-sensor-calibration

Edge Collective (2020). Radiohead LoRa Mesh Field Test. Test Report. Available at: https://edgecollective.io/notes/lincoln

Guerbaoui M, Afou YEL, Ed-Dahhak A, Lachhab A, Bouchikhi B (2013). PC-based automated drip irrigation system. International Journal of Engineering Science and Technology 5(1):221-225.

Hrisko J (2020). Capacitive Soil Moisture Sensor Calibration with Arduino. MakerPortal blog post. Available at: https://makersportal.com/blog/2020/5/26/capacitive-soil-moisture-calibration-with-arduino#:~:text=Calibration%20Procedure%3A&text=Fill%20container%20to%20200ml%20(or,refill%20the%20container%20to%20200ml

Kim Y, Evans RG (2009). Software design for wireless sensor-based site-specific irrigation. Computers and Electronics in Agriculture 66:159-165. Available at: http://smartfasal.in/wp/wp-content/uploads/2019/09/Software-design-for-wireless-sensor-based-site-specific-irrigation.pdf

Kim Y, Evans RG, Iversen WM (2008). Remote sensing and control of an irrigation system using a distributed wireless sensor network. IEEE Transactions on Instrumentation and Measurement, 57(7):1379-1387. Available at: https://pubag.nal.usda.gov/download/53900/pdf

Kisekka I, Aguilar J, Lamm F, Rogers D (2014). Using soil water and canopy temperature to improve irrigation scheduling for corn. Proceedings of the 2014 Irrigation Association Conference, Phoenix, Arizona. Available at: https://www.researchgate.net/publication/267865340_Using_Soil_Water_and_Canopy_Temperature_to_Improve_Irrigation_Scheduling_for_Corn

Lal R, Steward BA (2016). Soil-specific farming: Precision agriculture. Boca Raton, Florida, USA, CRC Press. Available at: https://amzn.to/3DpYDoH

Millán S, Casadesús J, Campillo C, Moñino MJ, Prieto MH (2019). Using soil moisture sensors for automated irrigation scheduling in a plum crop. Water 11(10):2061. Available at: https://www.mdpi.com/2073-4441/11/10/2061

Montagny S (2022). LoRa - LoRaWAN and the Internet of Things. Book and Course, Savoie Mont Blank University. Available at: https://tech-journal.semtech.com/lorawan-e-book-from-savoie-mont-blanc-university-now-available.

Munyaradzi M (2023, TBP). Adaptive control for precision irrigation scheduling in Zimbabwe. Dissertation, University of Zimbabwe, Faculty of Computer Engineering, Informatics, and Communications, Department of Computer Science.

Munyaradzi M, Hapanyengwi G, Masocha M, Mutandwa E, Raeth P, Nyambo B, Murwira A, Mashonjowa E (2022). Precision irrigation scheduling based on wireless soil moisture sensors to improve water use efficiency and yield for winter wheat in sub-saharan Africa. Hindawi, Advances in Agriculture. Available at: https://www.hindawi.com/journals/aag/2022/8820764

Munyaradzi M, Mashonjowa E, Rupere T, Mukute S, Nyambo BM, Chinyerutse M (2013). A low cost automatic irrigation controller driven by soil moisture sensors. International Journal of Agriculture Innovations and Research 2:1. Available at: https://www.researchgate.net/publication/260392710_A_Low_Cost_Automatic_Irrigation_Controller_Driven_by_Soil_Moisture_Sensors

Ncube B, Mupangwa W, French A (2018). Precision agriculture and food security in Africa. Book Chapter in Systems Analysis Approach for Complex Global Challenges pp. 159-178.

Nootropic Design (2018). LoRa mesh networking with simple arduino-based modules. Project Lab, open-access design exposition. Available at: https://nootropicdesign.com/projectlab/2018/10/20/lora-mesh-networking

Oliver M, Bishop T, Marchant B (2013). Precision agriculture for sustainability and environmental protection. Abingdon, Oxon: Routledge. Available at: https://amzn.to/3xmFOPr

Or D (2005). Soil Moisture sensors placement and interpretation for drip irrigation management in heterogeneous soils. 5th International Micro Irrigation Congress pp. 214-222.

Petousi I, Daliakopoulos IN, Matsoukas T, Zatos N, Mavrgiannis I, Manios T (2018). Development of an advanced precision drip irrigation system for tree crops. Poster, TerraEnVision, Barcelona, Spain. Available at: https://www.researchgate.net/publication/323005155_DRIP_Development_of_an_Advanced_Precision_Drip_Irrigation_System_for_Tree_Crops

Raeth PG (2023). A first shot at using LoRa for remote sensing. Project software and documentation repository, Remote-Soil-Moisture-Sensing.zip. Available at: https://github.com/SoothingMist/Scalable-Point-to-Point-LoRa-Sensor-Network.

Semtech (2023). What are LoRa and LoRaWAN. Semtech Documentation. Available at: https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/#:~:text=The%20name%2C%20LoRa%2C%20is%20a,areas%20(line%20of%20sight).

Shannon DK, Clay DE, Kitchen NR (2018). Precision agriculture basics. Madison, WI: American Society of Agronomy, Crop Science Society of America, Soil Science Society of America. Available at: https://amzn.to/3RKKVB1

Svedberg E (2019). Impact on yield and water productivity of wheat by access to irrigation scheduling technologies in Koga Irrigation Scheme. Uppsala University Publications, Ethiopia. Available at: https://uu.diva-portal.org/smash/get/diva2:1320170/FULLTEXT01.pdf

TutorialsPoint (2022). RESTful web services tutorial. Complete tutorial on the REST interface. Available at: https://www.tutorialspoint.com/restful/index.htm

Yosensi (2021). What is the real range of LoRa. Yosensi Documentation. Available at: https://yosensi.io/posts/what_is_the_real_range_of_lora/#:~:text=Introduction,as%20high%20as%20832%20km!

Zafar U, Arshad M, MasudCheema MJ, Ahmad R (2020). Sensor based drip Irrigation to enhance crop yield and water productivity in semi-arid climatic region of Pakistan. Pakistan Journal of Agricultural Sciences 57(5). Available at: https://www.researchgate.net/publication/343999188_Sensor_based_drip_irrigation_to_enhance_crop_yield_and_water_productivity_in_semi-arid_climatic_region_of_Pakistan

Zhang Q (2016). Precision agriculture technology for crop farming. Boca Raton, Florida, USA, CRC Press. Available at: https://amzn.to/3UdpGd0