LORA FLOOD-MESSAGING FOR SENSOR-DATA TRANSPORT

Peter G. Raeth
Senior Research Engineer, USA
peter_raeth@ameritech.net

A paper from the Proceedings of the
16th International Conference on Precision Agriculture
21-24 July 2024, Manhattan, Kansas, United States

Abstract. *Precision agriculture assumes the ability to place and monitor sensors. Remote monitoring is often employed as a means of alleviating tedious manual data gathering and recording. For remote monitoring to work there has to be some automated means of reading sensor values and transmitting them to a basestation, someplace where the data is recorded, analyzed, and employed. If the data are recorded and analyzed at the point of sensing, some means is still required to send the results to wherever they are needed. Well-developed nations have cost-effective and reliable internet and cellular communication systems. Nations in development, and remote areas in all nations, may not have such systems. In any case, both systems are operated by third-parties that require users to pay setup costs continuing fees. Another disadvantage is the control third-parties have over access and operation. An alternative is LoRa radio broadcast. LoRa operates on unlicensed frequencies. The equipment required is inexpensive and readily available world-wide. There are no continuing fees. It is meant for and legally-constrained to low power, message-rate, bandwidth, and message-size applications. This proof-of-concept demonstrates a LoRa flood-messaging network that avoids the complexity and expense of LoRaWAN and third-party services. A wireless network is formed through which messages flow from sensors attached to microcontrollers, through transmission relays (to account for distance, terrain variability, and other obstructions), to a basestation connected to some external system. The very same LoRa hardware is used for all network components. Relays all use the same software. The same is true of basestations. Baseline software for sensor nodes adds programming based on the types and numbers of sensors attached. Each sensor, relay, and basestation has its own unique assigned network address. Single-value sensor data transport is demonstrated, as well as the transport of complex sensor data such as images. Message-integrity checks are included. Indoor and outdoor networks are demonstrated. Documentation and software for the successful proof-of-concept discussed in this paper are posted for public download.*

Keywords. *LoRa, peer-to-peer, IoT, precision agriculture, precision irrigation, mesh networking, flood messaging, ad hoc networks, wireless sensor networks, remote sensing*

The authors are solely responsible for the content of this paper, which is not a refereed publication. Citation of this work should state that it is from the Proceedings of the 16th International Conference on Precision Agriculture. EXAMPLE: Last Name, A. B. & Coauthor, C. D. (2024). Title of paper. In Proceedings of the 16th International Conference on Precision Agriculture (unpaginated, online). Monticello, IL: International Society of Precision Agriculture.

Introduction

Precision agriculture combines agricultural knowledge with other fields such as physics, engineering, geography, remote sensing, automation, and meteorology. A true team effort is necessary. The goal is to maintain or improve crop value while using fewer resources such as land, pesticide, water, and fertilizer. It does so while retaining nutrition and without resorting to the risks of genetic mutation. Given the vastness, isolation, and variable terrain of many farms, remote sensing is often critical. Moving data from sensor locations to where it is processed, displayed, and employed is necessary to avoid manual monitoring and collection. Considering the flexible nature of productive farming, an approach using relatively-inexpensive equipment that is easily modified and placed is needed for sensor data communication.

Sensor data communication is often performed using two major approaches: cellular telecommunication and internet connection. Both are effective and have their place given the vast variation in circumstances and application requirements. However, both involve third-party services and their continuing fees, besides the necessary equipment. There is also the assumption of cost-effective reliable existence. Such an assumption often does not hold true in developing nations and the rural and remote areas of all nations. This paper explores an alternative, LoRa (Long Range) peer-to-peer (P2P) radio broadcasts, a wireless approach to communicating digital data via unlicensed frequencies. LoRa does not require third-party services while still enabling relatively long-range communication (up to two miles over flat terrain, depending on the transceiver/antenna employed). Relay transceivers provide range extension despite obstructions and distance.

LoRa P2P stands in contrast with LoRaWAN, a more expensive and complex technology. Both operate on the same frequencies but employ different protocols. They are not directly compatible (although some transceivers can operate in both modes). Within the layers of the Open Systems Interconnection (OSI) Reference Model (see Saylor Academy for a overview) LoRa P2P operates at the physical and data link levels. LoRaWAN provides for the network and above levels. Another way of looking at it is that LoRa provides the wireless modulation that creates a communication link. LoRaWAN is the communication protocol, equipment, and system architecture for building a large network. Montagny has offered an excellent book on LoRa/LoRaWAN should you want to study the matter in more detail.

It is this author's assertion that many applications do not require LoRaWAN. By connecting LoRa transceivers with programmable devices such as microcontrollers or embeddable computers, it is possible to build a P2P network using LoRa that does not involve the expense and complexity of LoRaWAN. For large-scale networks, LoRaWAN may well be a better answer but for small or medium networks LoRaWAN can be overkill. LoRaWAN involves servers, gateways, and other expensive equipment, whereas LoRa P2P uses relatively inexpensive programmable LoRa transceiver nodes. We will see that only the nodes themselves are used to build the network. The very same nodes are employed throughout, with some variation in software depending on the role a given node plays and the sensors employed.

The rest of this paper will explore the specifics of this proof-of-concept. We will address the hardware and software required to build a network. Then we will show the results of testing and an initial thrust at solar-powered nodes and all-weather installation. A conclusion and thoughts on next steps finishes the paper. Citations are given where warranted. All software developed for this effort and additional documentation are available for public download: https://github.com/SoothingMist/Scalable-Point-to-Point-LoRa-Sensor-Network.

Hardware

This present effort relies on the foundation laid by Branch, Li, and Zhao. They developed a LoRa point-to-point flood messaging system for underground RFID. We can expand beyond the RFID case if we recall that, fundamentally, LoRa is designed to move 250-byte data packets at low rates between source and destination. Whatever is in those data packets is a matter of determination and interpretation. Thus, one can consider sending the value derived from a sensor that produces a single data value on demand. One can also pack multiple data values if the node services more than one sensor.

Some sensors produce complex data objects. Common cameras are the most basic example. They produce a one-layer or three-layer data object (image). Each layer has single-value data spread across a grid. black-and-white and grayscale cameras produce a one layer image. A red/green/blue (RGB) camera produces a three-layer image. RGB+NIR (near infrared) produces a four-layer image. There are imagers that produce a number of layers in the thousands. Thus the use of satellite imagery in agricultural applications. (See Mathenge, Sonneveld, Broerse for more on this topic.) It is not unseemly to imagine sending RGB or RGB+NIR data across a LoRa peer-to-peer network. We demonstrate that capability in this paper.

To build a LoRa peer-to-peer network, the author started with two basic modules: a microcontroller development board and a LoRa peer-to-peer transceiver. These are depicted in Figure 1. Together, these cost $US50-60 delivered.

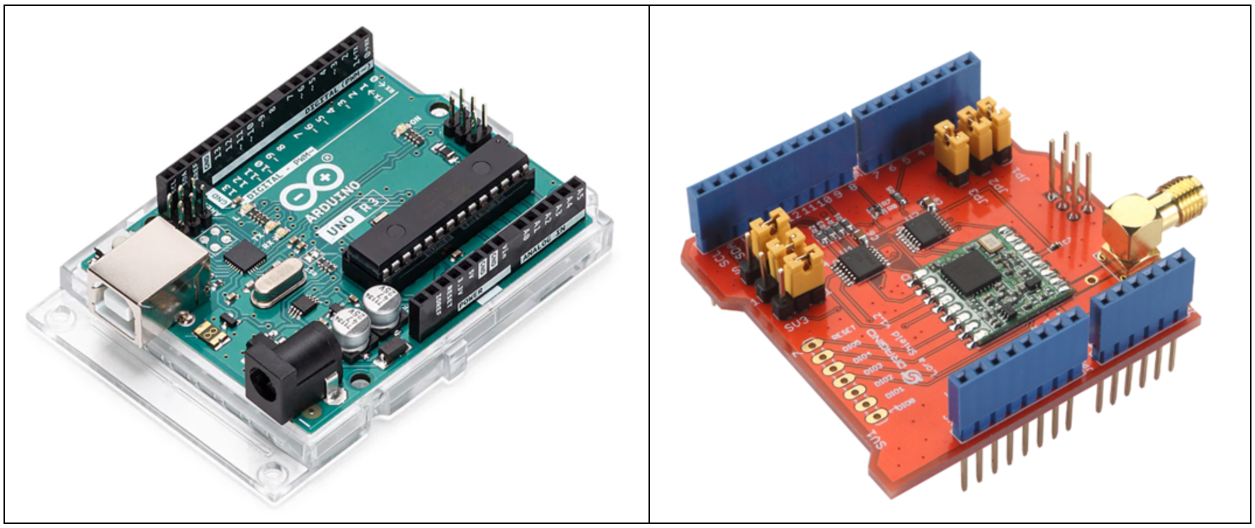

Figure 1. Microcontroller development board and LoRa peer-to-peer transceiver.

The microcontroller development board is the Arduino Uno R3. The transceiver is the Dragino LoRa Transceiver. Both are readily available internationally and are well documented. The transceiver can be configured for the frequency mandated by a given country, although care must be taken in selecting an appropriate hardware/firmware version. The transceiver plugs directly into the top of the microcontroller board. No other connection is required. A quality antenna comes with the transceiver which screws into the socket shown on the right of the device.

This unit forms the heart of the network. However, with the microcontroller/transceiver selected, a bit of hardware expansion was needed for complex sensors such as cameras. A Pixy v2.1 camera (Figure 2) plugs directly into the microcontroller board. However, it uses the same pins/connections as does the transceiver. Thus, there is no electronic compatibility. The three-way unit cannot work.

One solution is to build a programmable USB hub since that is the way the microcontroller board connects to its serial port. For that purpose, a Raspberry Pi embeddable computer (Figure 2) was used. One microcontroller hosts the camera and another hosts the transceiver. Data collected by the camera is passed to the transceiver through the USB Hub for processing and transmission.

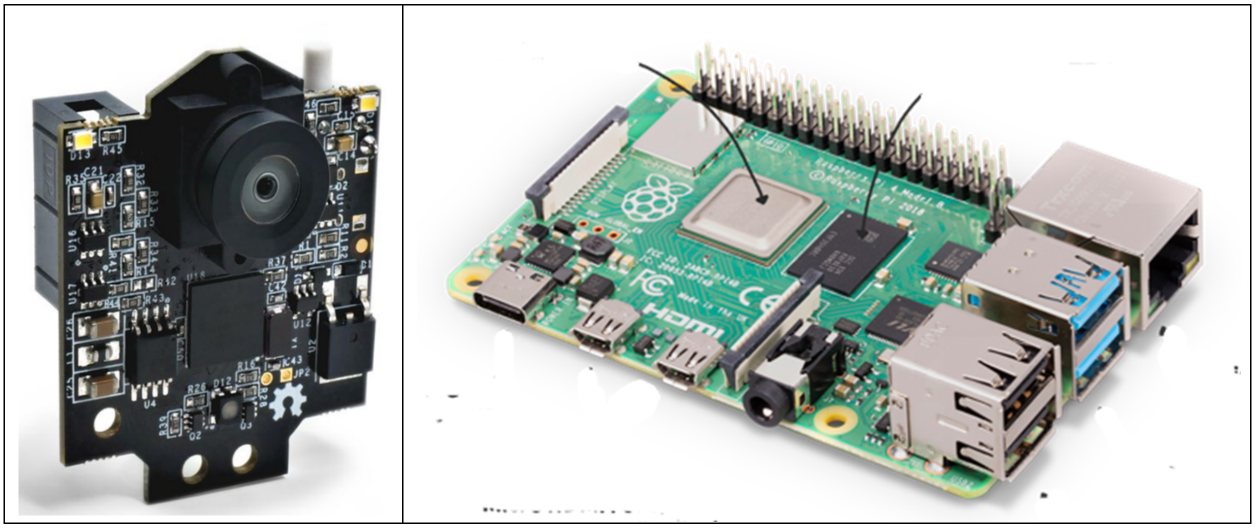

Figure 2. Pixy v2.1 camera and Raspberry Pi embeddable single-board computer.

Why bother with a camera? It may be necessary to remotely observe some infrastructure, such as an on-farm weather station. Also, there are many types of 'cameras'. For instance, an NIR+RGB camera has four layers of data, to include infrared. These can be used to estimate the dryness of a crop (Steven). Even with just a common inexpensive RGB camera, it is possible to arrive at a general estimate (Rabatel, Gorretta, and Labbe). Each node can be programmed to operated as a relay, sensor host, or basestation. For most sensors, their output connects to a microcontroller analog pin. Software interprets the digital value according to the sensor type and calibration.

Figure 3 Illustrates how the various components go together in a network. The number of sensor and relay nodes can be expanded as needed. More than one basestation can be employed.

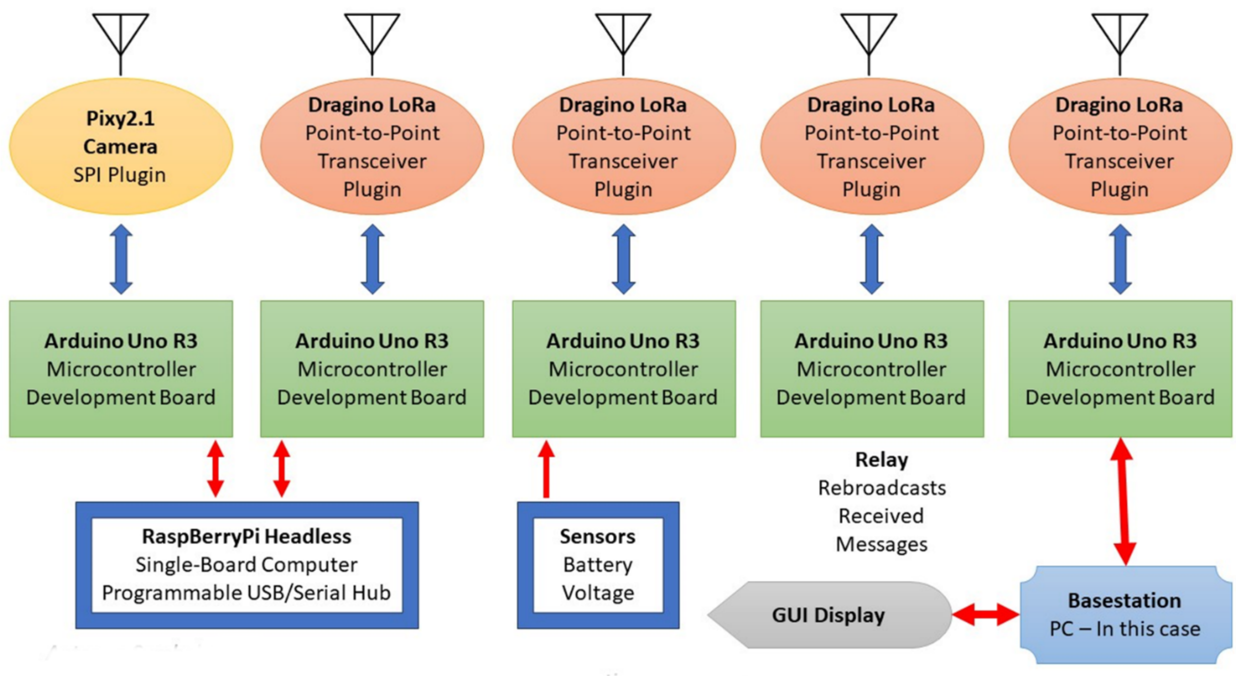

Figure 3. Network of LoRa peer-to-peer nodes.

The arrangement illustrated in Figure 3 does not forestall connection to third-party services such as those on the internet or via telecommuncation. That is a matter of application requirements and specification satisfied by appropriate programming and equipment.

Software

Within the physical network, a flow of data is maintained. Sensors gather and transmit data, relays receive and forward data, basestations receive, process, employ, and display data. For an overview of this process, see Figure 4.

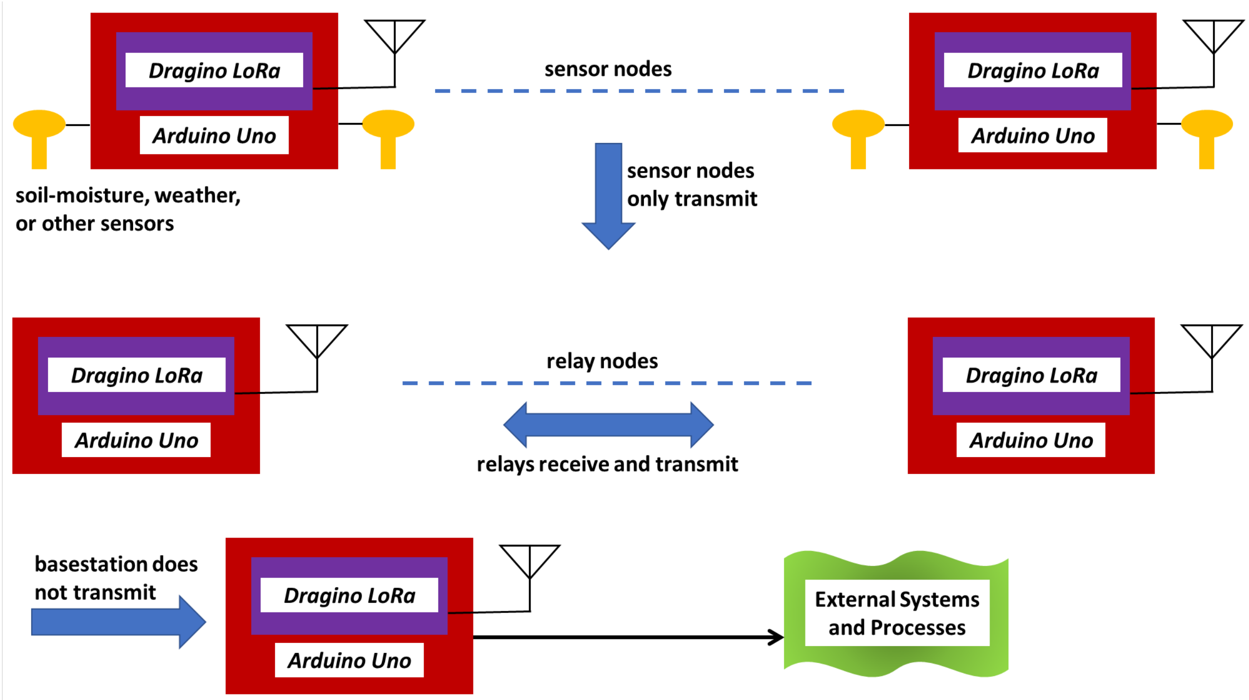

Figure 4. Flow of data within a LoRa P2P flood-messaging network.

Data flowing in the network is encapsulated within messages of no more than 250 bytes. There are certain message types, depending on the type of data being sent. For example, there may be camera data, text notifications, or single-value data. There may also be commands to the network itself. Each message starts with the standard header shown in ***Table 1***.

Table 1. Standard message header

Byte	Meaning
0	Message Bytes
1	CRC High Byte
2	CRC Low Byte
3	System ID
4	Source Node ID
5	Destination Node ID
6	Source Message ID - High Byte
7	Source Message ID - Low Byte
8	Message Type
9	Source Sensor ID
10	Number of Rebroadcasts (ttl, time to live)

The number of times a message is rebroadcast by any relay is controlled by Byte 10. Each time a message is rebroadcast, this counter within the message is reduced. Notice that relays and basestations do not originate messages. Only sensor nodes originate messages. Lets have a look now at pseudo-code for the software running on each node type.

Sensor Nodes: The logic for sensor nodes is really quite simple. Sensor readings are periodically taken and transmitted, as shown in Figure 5.

```
while true do
    for each sensor associated with this node
        read sensor value
        adjust sensor value using calibration factor
        increment sensor's message sequence
        add time to live
        assign message header
        assign message data
        transmit message
        wait an amount of time appropriate to the application
```

Figure 5. Pseudo-code representation of logic employed by sensor nodes.

Relay Nodes: Software for all relays is always the same. The underlying paradigm is exposed in Figure 6. 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.

```
while true do
    while channel is idle
        monitor channel for message
    upon receipt of message
        check message CRC
        if CRC failed
            discard messaged
        else
            if messageID <= sensorList[sensorID]
                discard message
            else
                if message.type eq Reset
                    sensorList[sensorID] = 0
                else
                    sensorList[sensorID] = messageID
                message.ttl = message.ttl – 1
                if message.ttl <= 0
                    discard message
                else
                    update message data fields as needed
                    adjust CRC
                    wait exponentially distributed random amount of time, mean 100 ms
                    transmit message
```

Figure 6. Pseudo-code representation of logic employed by relay nodes.

Basestation Node: Basestation software is very similar to that of relays. As currently envisioned, the basestation is a destination node. All others are either sensor nodes which transmit data or relays that forward messages. Figure 7 uses pseudo-code to express the underlying logic. The basestation itself decodes the message and uses its contents for the intended purpose.

```
while true do
    while channel is idle
        monitor channel for message
    upon receipt of message
        check message CRC
        if CRC failed
            discard messaged
        else
            if destinationID == basestationID
                forward message to basestation
```

Figure 7. Pseudo-code for basestation node logic.

Camera Nodes: The node logic offered so far works very well for sensors that generate only a single value. A text message is fine for that. To go beyond, one needs a way to compact multiple values into a single message. That is easily done since more than one 16-bit (2 bytes) numerical value can be inserted into a single LoRa message. (Note from ***Table 1*** how the standard message header contains two-byte data.) For images, it is another story. Multiple messages are needed to convey the entire data object. As the dimensions of the image grow, so does the number of messages required. The present implementation sends images row-by-row. The basestation reconstructs the image as messages arrive. Rows are segmented into pixel components so that a number of pixels is placed into a message according to the number of layers in each pixel. The software automatically adjusts to the number of layers, rows, and columns. Figure 8 shows an example.

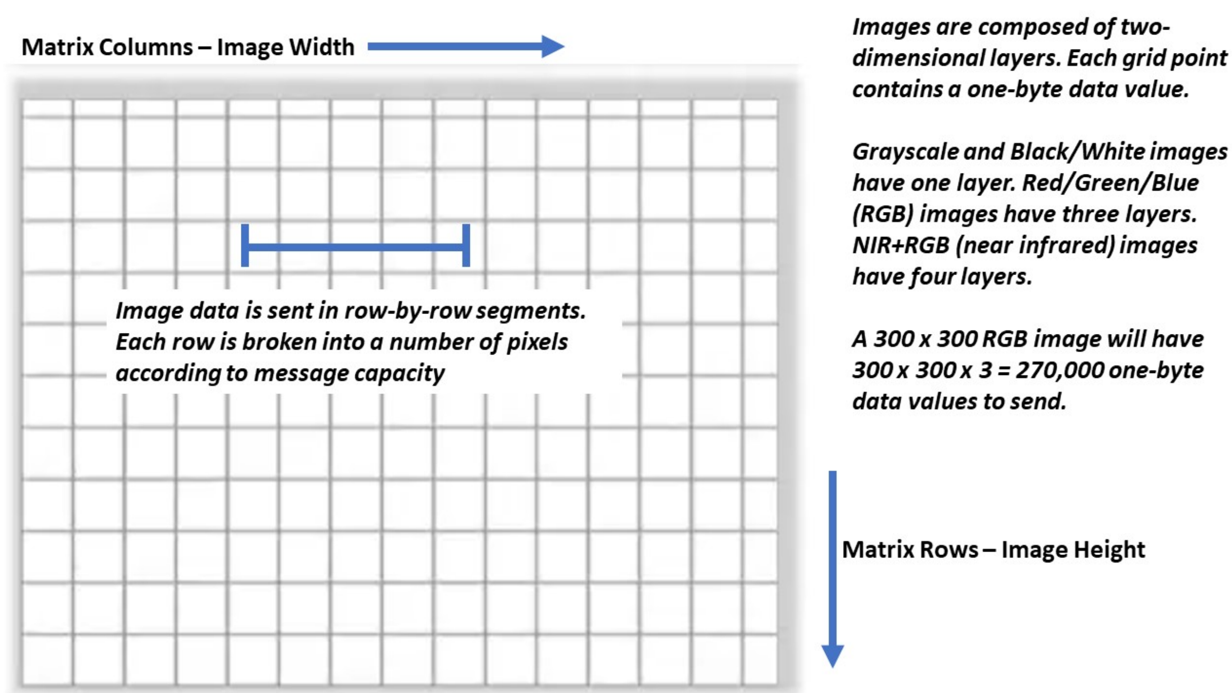

Figure 8. Illustration of image sent row-segment by row-segment.

Testing

To test this concept of LoRa P2P Flood-Messaging, the author constructed indoor and outdoor networks. The indoor network contained a single-value sensor node, a camera node, a relay node, and a basestation node. The outdoor network contained a single-value sensor node that sent the voltage of the battery and the voltage of the solar-cell that recharged the battery. Figure 9 explains the GUI display produced by the basestation.

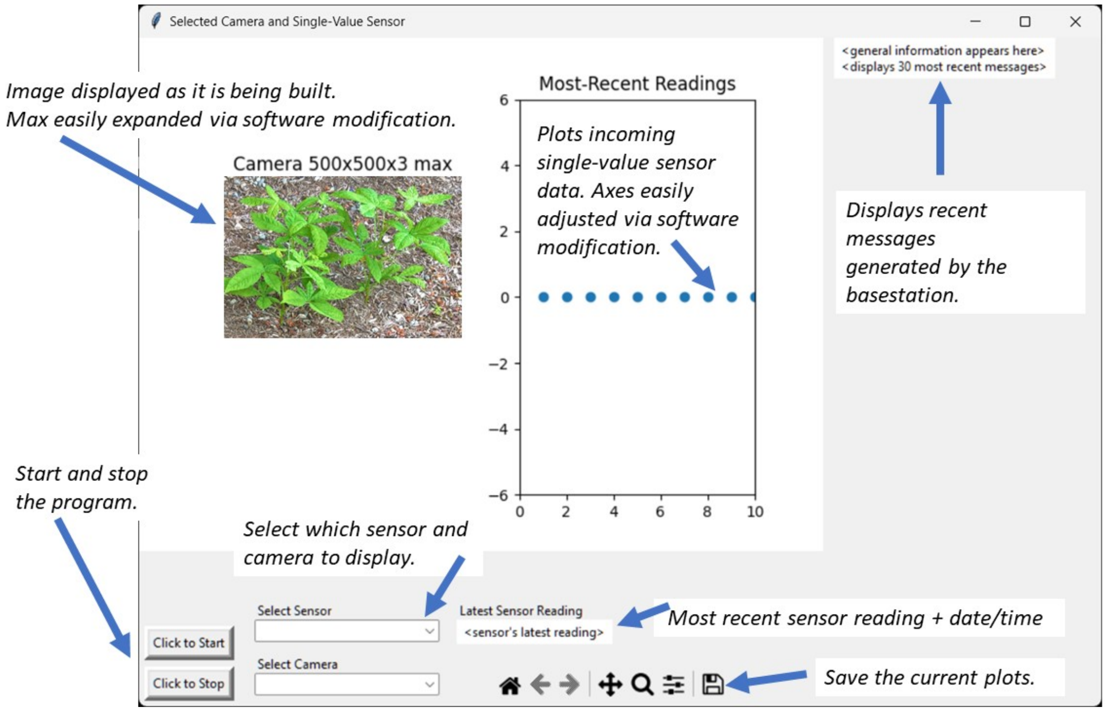

Figure 9. Explanation of the basestation's GUI display.

The indoor test had heavy electromagnetic interference from surrounding equipment. About 1 in 800 messages were lost when no relay was employed. When a relay was employed, no messages were lost. This was due to the duplication of message traffic produced by the relay.

The outdoor test used a weatherproof enclosure. Wires for external equipment entered through a hole in the enclosure that was protected by a common grommet. The microcontroller/transceiver was inside the enclosure, as was the antenna. Distance was about one block. Reception results were poor until a relay was placed in the house's sunroom which faced the sensor node's location on the other side of the house from the basestation. Then there were no lost messages, when the node was operating. Dragino documentation claims up to two miles in flat unobstructed terrain when a quality antenna is extended in open air.

A word is needed to explain "when the node was operating" during the outdoor test. Beale's method #1 was applied, with the substitution of a rain-proof solar-cell and a 2000mAh Lithium-Ion battery. The solar cell was mounted on the side of an outbuilding facing the morning sun. The node was placed on the ground. The basestation records all data it receives. A table lamp was used to energize the solar cell, which fully charged the battery. Then the unit was mounted outdoors. Sunny days occurred at the start of the trial. Temperatures ranged from 22F – 40F. Beyond a bit of morning frost, there was some rain but no ice or snow. Figure 10 shows the plot of data recorded by the basestation. This cycle recurred during relatively sunny days.

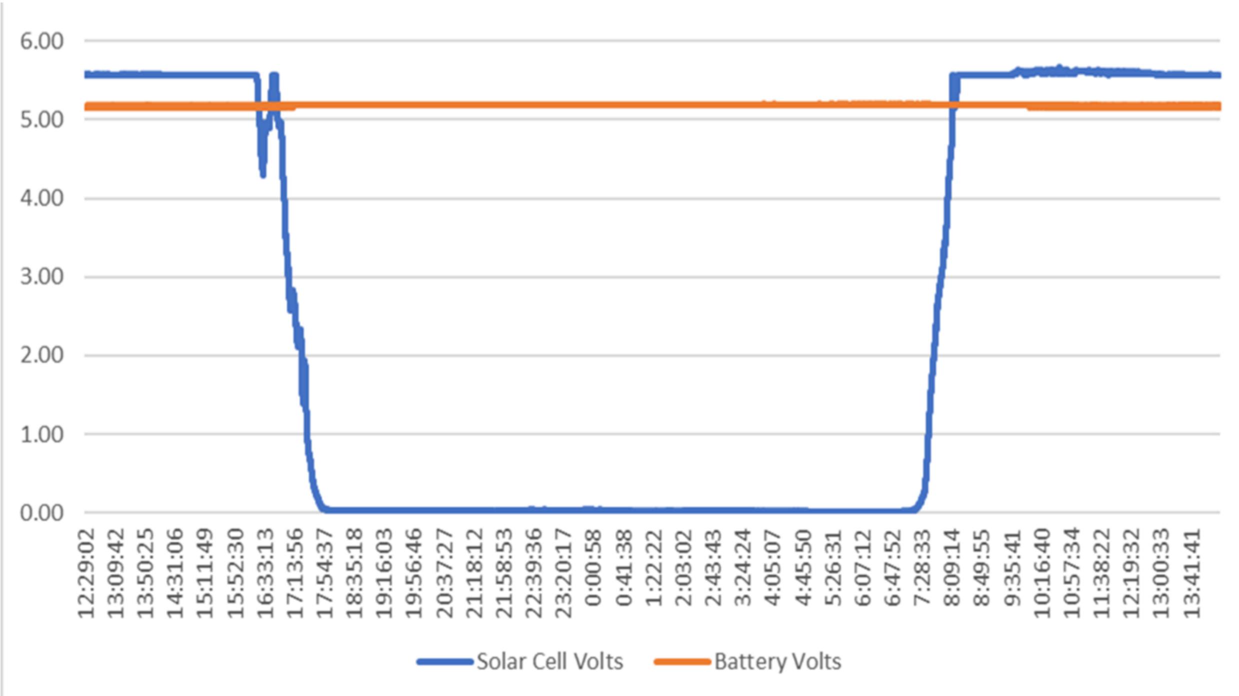

Figure 10. Plot of solar-cell and battery data that recurred during relatively sunny days.

As expected, the voltage delivered by the battery remained nearly constant while the voltage delivered by the solar cell dropped during evening hours. Solar-cell voltage was restored as the sun rose again. This was the pattern during sunny days.

However, there came several days of very overcast skies. It was not like night but the sun did not shine well enough to keep the battery charged. Performance was very erratic during that time. The solar cell would charge the battery enough to briefly operate the node but then the battery would quickly drain because of insufficient charge. Finally, sunny days came again, the battery was fully charged, and the sunny-days cycle continued. (One would expect ice or snow to obscure the solar cell even on sunny days. But that did not occur during this test.)

Clearly, the 2000mAh battery was insufficient to ensure reliable operation during repeated days of overcast sky. A larger battery would be needed for that. To charge a larger battery, one should also investigate larger solar cells and different charge controllers. One also should consider the number of charge/recharge cycles the battery can tolerate. Another thrust is to determine what could be turned off on the microcontroller board. In this regard, a separate experiment showed that, if sensor nodes can be powered and depowered on schedule to deliver readings every six hours, a 2000mAh lithium battery could last up to three years without recharging. All-Weather remote-powering is another important area of work if 24/7/366 operation is required.

Discussion and Next Steps

There are many approaches and equipment families that can achieve LoRa P2P in sensing/processing/control applications. Suhermanto et al demonstrate an application for monitoring and controlling a DC motor. Their approach could lead to lower costs but it does not use the same equipment for all transceiver nodes. They do keep the interface with external systems separate from the network itself. The present proof-of-concept uses a PC as the external system. Theirs communicates with the cloud. In our case, a cloud connection could be made by the PC via its internet connection. Integrating their ideas, nodes could be expanded to communicate directly with third-party services.

Encryption was not included in this proof-of-concept because of the memory limitations of the microcontroller board. Sanjay, Rao, and Amith do employ encryption but use a PC as part of the process because their chosen microcontroller could not handle the entire algorithm. They mention error correction but do not implement it. As implemented, this present proof-of-concept uses error detection and rejects messages where CRC does not match. There is no attempt to perform error correction. Their effort at secure communications is noteworthy and can be useful in certain applications.

Berto, Napoletano, and Savi make an interesting comment, *“One of the strongest limitations of LoRaWAN is the adopted topology, where only direct single-hop communication is allowed between end devices and gateways. Even though this configuration is suitable for many applications, in some cases (e.g., when data must be gathered/exchanged from/in difficult-to-access areas) it is far from being the optimal solution.”* This argues in favor of their mesh network and flood-messaging. In this present work, relay nodes enable multi-hop of messages. Similarly too, they use inexpensive microcontroller/transceiver nodes. They demonstrate the value of their design but do not include features such as encryption/decryption and error detection. Given the hardware employed, that is understandable. Their goal was to demonstrate mesh networking of low cost and power consumption.

The present proof-of-concept uses Dragino’s Lora Shield. Dragino’s Lora Shield is NOT Dragino’s LA66 LoRaWAN shield. That is something different. The software developed in this project will not run on that shield. At the start of 2024 the author found out that Dragino’s Lora Shield is at its end-of-life, is scheduled for obsolescence, and will be discontinued. It is still available through many sellers. Their new shield (LA66 LoRaWAN), according to its documentation, is capable of both peer-to-peer and LoRaWAN. Message handling in this proof-of-concept’s software is kept seperate from code that interacts with LoRa. Thus, transition should be achievable, although the new shield uses a completely different software library. It will be interesting to see if the new module LoRa P2P is compatible with the old module. The new and old modules are mechanically and electrically plug-compatible. This author prefers LoRa modules and other components that plug directly into the microcontroller board since that avoids a rats’ nest of wires and thus makes the result more reliable.

As one considers expanded applications, one is led to wonder why all node types could not also act as relays and be capable of transmit/receive. For instance, an operational node may be used to trigger some device’s activity, pumping water for instance. Transmit/Receive nodes would certainly present a real opportunity. Of concern is that the microcontroller board presently being used is already near its memory limits. One would need something compatible but with larger memory. In this case, an Arduino Uno R3 microcontroller board was being used. The Uno R4 does not solve this issue. One needs to move to the Mega 2560 R3 for its expanded memory and plug-compatibility. This microcontroller board costs about $US55 delivered, a cost increase that must be taken into account.

I2C communication should be further explored as a means of linking two microcontroller boards. If successful, that would eliminate the need for a programmable USB hub for camera sensors. According to Wu, *“I2C is a two-wire serial communication protocol using a serial data line (SDA) and a serial clock line (SCL). The protocol supports multiple target devices on a communication bus and can also support multiple controllers that send and receive commands and data. Communication is sent in byte packets with a unique address for each target device.”* The microcontroller board at hand is capable of interboard I2C communication.

Conclusion

The goal of this work has been to show LoRa point-to-point communication is capable of providing remote access to sensors delivering single-value, multi-value, and multi-layered data. That goal has been achieved. The realities of field-testing have been explored and specifics have been identified. The value to precision agriculture is that sensor data can be remotely accessed without the expense and complexity of LoRaWAN and without the continuing fees of third-party services, A further value is the flexibility and relatively low-cost of this technology so that even small-scale farmers can use it.

References

Beal A (2022). Three Ways to Solar Power an Arduino. Footprint Hero, Instructional Article, Available from https://footprinthero.com/how-to-solar-power-an-arduino

Berto R, Napoletano P, Savi M (2021, June). A LoRa-Based Mesh Network for Peer-to-Peer Long-Range Communication. MPDI Sensors, 21(13), 4314, Available from https://www.researchgate.net/publication/352707672_A_LoRa-Based_Mesh_Network_for_Peer-to-Peer_Long-Range_Communication

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. Available from: https://www.mdpi.com/2673-4001/1/2/6

Mathenge M, Sonneveld BGJS, Broerse JEW (2022). Application of GIS in Agriculture in Promoting Evidence-Informed Decision Making for Improving Agriculture Sustainability: A Systematic Review. MDPI, Sustainability, 14(16):9974. Available from https://www.mdpi.com/2071-1050/14/16/9974

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

Rabatel G, Gorretta N, Labbe S (2011) Getting NDVI spectral bands from a single standard RGB digital camera: A methodological approach. In: Lozano, J.A., Gámez, J.A., Moreno, J.A. (eds) Advances in Artificial Intelligence. CAEPIA 2011. Lecture Notes in Computer Science(), vol 7023. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-25274-7_34, Available from https://link.springer.com/chapter/10.1007/978-3-642-25274-7_34

Sanjay GK, Rao SBR, Amith G (2022). Encrypted P2P Communication using LoRa Waves. TechRxiv, Available from https://www.researchgate.net/publication/364102003_Encrypted_P2P_Communication_using_LoRa_waves

Saylor Academy (2020, Oct 1). The OSI Reference Model. CS402 course material. Available from https://learn.saylor.org/mod/page/view.php?id=67555

Steven MD (1998) The sensitivity of the OSAVI vegetation index to observational parameters. Remote Sensing of Environment, 63, 49-60, Available From https://www.researchgate.net/publication/229367131_The_Sensitivity_of_the_OSAVI_Vegetation_Index_to_Observational_Parameters

Suhermanto DANK, Aribowo W, Shehadeh HA, Rahmadian R, Widyartono M, Wardani AL, Hermawan AC (2024). Monitoring DC Motor Based on LoRa and IOT. Journal of Robotics and Control, 5/1, Available from https://www.researchgate.net/publication/377242172_Monitoring_DC_Motor_Based_on_LoRa_and_IOT

Wu J (2022). A Basic Guide to I2C. Texas Instruments, Application Note, Available from https://www.ti.com/lit/an/sbaa565/sbaa565.pdf?ts=1706766045707&ref_url=https%253A%252F%252Fwww.google.com%252F#:~:text=I2C%20is%20a%20two%2Dwire,and%20receive%20commands%20and%20data.

Picture Credits

- https://commons.wikimedia.org/wiki/File:Antenna_schematic_symbol.svg
- https://www.raspberrypi.com/products/raspberry-pi-4-model-b
- https://pixycam.com/2021/05/20/introducing-pixy-2-1
- https://store-usa.arduino.cc/products/arduino-uno-rev3
- https://www.dragino.com/products/lora/item/102-lora-shield.html
- https://www.101planners.com/borders/graph-paper/#12_Inch_Graph_Paper
- https://gardening.stackexchange.com/questions/27597/does-anyone-know-what-plant-is-this-see-picture-below