Spring 3-2021

Development of an Internet of Things (IoT) Enabled Novel Wireless Multi Sensor Network for Infield Crop Monitoring

Nipuna Chamara
University of Nebraska-Lincoln, nabeysingheherathm2@huskers.unl.edu

DEVELOPMENT OF AN INTERNET OF THINGS (IOT) ENABLED NOVEL WIRELESS MULTI SENSOR NETWORK FOR INFIELD CROP MONITORING

by

Nipuna Chamara Abeysinghe Herath Mudiyanselage

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfilment of Requirements

For the Degree of Master of Science

Major: Agricultural and Biological Systems Engineering

Under the Supervision of Professor Yufeng Ge

Lincoln, Nebraska

March 2021

DEVELOPMENT OF AN INTERNET OF THINGS (IOT) ENABLED NOVEL WIRELESS MULTI SENSOR NETWORK FOR INFIELD CROP MONITORING VEGETATION INDEX MONITORING

Nipuna Chamara Abeysinghe Herath Mudiyanselage, M.S.

University of Nebraska, 2021

Advisor Yufeng Ge

Multispectral imaging systems on satellite, aerial, and ground platforms are used commonly to monitor in-field crops in precision agriculture by farmers and researchers. Limited spatial and temporal resolution and weather dependence of the data collection are two main disadvantages of these methods. In-field sensor networks can continuously monitor environmental and plant physiological parameters by leveraging low-power computation and long-range communication technologies. We built and tested a novel sensor network equipped with soil moisture, multispectral and RGB imaging sensors in an experimental soybean field at Eastern Nebraska Research and Extension Center, NE, USA. 10 down-looking and 1 up-looking sensor node were set up at the experimental site in the 2020 growing season. Each down-looking sensor node utilized an 18-channel multispectral sensor, a soil moisture sensor, and an RGB imager to collect canopy reflectance, soil volumetric water content (VWC), and canopy images every 20 minutes. The up-looking sensor node measured the solar radiation using a multi-spectral sensor every 10 minutes. The setup allowed us to calculate the spectral reflectance of the soybean canopy under changing weather conditions. The sensor nodes were solar-powered and integrated into a Low Power, wide area network (LoRaWAN) through a LoRa gateway, which was connected to the internet via Wi-Fi. Captured images were

saved on SD cards while other parameters were uploaded to cloud data storage for real-time processing and visualization. The result shows that the sensor network can plot canopy Normalized difference vegetation index (NDVI) and soil VWC continuously throughout the growing season. NDVI values of the irrigated and rainfed soybean plots showed a significant difference. The coefficient of determination or the R squared value was 0.8701 between the GreenSeeker Handheld Crop Sensor and the IoT enabled sensor node NDVI value. This research verified that NDVI value is not constant throughout the day. Daily NDVI variation has two peaks between 10.00 -11.00 am and 1:00 - 2:00 pm. This sensor network could help users to estimate the crop growth parameters and irrigation requirements in a real-time fashion. Further, the diurnal NDVI tracking with an in-field NDVI sensor node has the potential to improve the NDVI value-based variable-rate fertigation unit efficiency improvement.

Keywords: IoT, multispectral sensor, vegetation indices, NDVI, spectral reflectance sensor, PRI.

ACKNOWLEDGMENTS

First, I would like to express my sincere gratitude to my advisors Dr. Yufeng Ge for the continuous support, guidance, and motivation throughout my master's program while being a great teacher and a role model.

Also, I would like to acknowledge my co-advisors Dr. Geng (Frank) Bai and Dr. Kuan Zhang. Their insightful comments and vital advice improved my research outcomes.

Next, I would like to thank my friends Dr. Nuwan Wijewardane and Dr. Abbas Atefi for their help in my research and for giving me good ideas and suggestions. They made my academic and personal life successful in the last two years in Lincoln, Nebraska.

This thesis would not have been possible without my wife, Chathurika. I want to thank her for her loving support and for helping me with my research on various levels. Also, I need to thank my son Thivain for making life enjoyable and energetic. Also, I would like to express my very profound gratitude to my parents and two brothers for their continuous encouragement throughout my years of study and my life in general.

Finally, I need to express my special gratitude to Dr. D. A. Nimal Dharmasena, Dr. Premalal Kuruppuarachchi, and Dr. K.H. Sarananda for their forward-looking career guidance advice to motivate me to do postgraduate studies.

Finally, I wish to express my appreciation to all the people who have supported this research in any way.

TABLE OF CONTENTS

Chapter 1 GENERAL INTRODUCTION 1

1.1. Introduction to Internet of Things (IoT) 1

1.2. IoT in Agriculture (Crop Production) 2

1.3. Overview of Current IoT Systems 4

1.3.1. Existing Agricultural IoT System Architecture 4

1.4 IoT System Layers 7

1.4.1 Perception Layer 8

1.4.2 Network Layer 9

1.4.3 Middleware layer 10

1.4.4 Application Layer 10

1.4.5 Business Layer 11

1.5 Monitoring Crop Canopy Vegetation (Pigment) Indices 11

1.6 Conclusion 12

Chapter 2 DEVELOPMENT AND FIELD TESTING OF THE NOVEL IOT SENSOR NETWORK FOR INTEGRATED SOIL, CROP, AND ENVIRONMENT SENSING 14

2.1 Introduction 14

2.2 Hardware Selection and System Design 14

2.2.1 Spectroscopy Sensor Selection 15

2.2.2 Spectroscopy Sensors Calibration..... 18

2.2.3 Reflectance Calculation 19

2.2.4 Vegetation Indices Calculation 19

2.2.5 RGB Camera Selection 19

2.2.6. Soil Moisture Sensor 21

2.2.7 Integrated Sensor Node Design Analysis 21

2.2.8 Sensor Node Physical Arrangement 26

2.3 Wireless Sensor Network Design 26

2.3.1 The Quality of Service (QoS) 28

2.4 IoT Analytics Platform 29

2.4.1 ThingSpeak Setup, Features, and Specifications 30

2.5 IoT Enabled Sensor Network Field Testing 34

2.5.1 Location 34

2.5.2 Treatments and Measurements 34

2.5.3 Ground Truth Data Collection 35

2.6 Technical Challenges Faced 36

2.7 Conclusion 37

Chapter 3 RESULTS AND DISCUSSION 38

3.1 Introduction 38

3.2 LoRaWAN Network Performance Evaluation 38

3.3 Physical Sensor Node Design Evaluation and Novel Node Design Proposal 42

3.4 Spectroscopic Sensor Validation 43

3.4.1 In-house Testing 43

3.4.2 Spectroscopic Sensor Accuracy Validation with LI-190R Quantum Sensor 45

3.5 Diurnal and Seasonal NDVI Variation 48

3.6 The capability of IoT Sensor Network to Distinguish the Two Treatments. 51

3.7 Manual NDVI Value vs Sensor Node NDVI Comparison 52

3.8 Competitive Advantages of the Novel Sensor Network Over Commercial Solutions 54

Chapter 4 GENERAL CONCLUSIONS AND WAY FORWARD 58

References 61

Appendix 70

LIST OF FIGURES

Fig. 1.1 Schematic diagram of the architecture of the FarmBeats IoT system.....5

Fig. 1.2 System components of the Internet of Things (IoT) enabled platform for the research in precision agriculture and ecological monitoring from an implementation viewpoint (Popovic, et al., 2017) 6

Fig. 1.3 Perception layer device architecture..... 8

Fig. 2.1 Spark fun triad spectroscopy sensor board..... 15

Fig. 2.2 AS7265x photodiode arrays sensitive wavelengths with reference letter 16

Fig. 2.3 The up-looking senor node (A) and the down-looking sensor node (B) of the IoT sensor network. The components of the sensor nodes included (1) Lambertian diffuser (2) Circuit enclosure of the sensor node (3) LoRa Antenna (4) Supporting pole (5) Supporting pole for down looking sensor (6) Solar panel (7) Down looking sensor node (8) Sensor calibration Lambertian surface (9) Soil water content sensor cable 17

Fig. 2.4 Average Field of View of the Spectroscopy Sensor (AG, 2018) 18

Fig. 2.5 ArduCam camera shield RGB camera with a five Mega Pixel CMOS (Complementary Metal Oxide Semiconductor) imaging sensor (OV5642, Omni Vision Technologies, Inc., California, USA) 20

Fig. 2.6 IoT sensor node power supply lines, data flow lines, and control signal lines visualization..... 24

Fig. 2.7 Schematic diagram of the IoT sensor node structure with main components A) detailed view of node electronics, B) detailed structural arrangements 25

Fig. 2.8 Architecture and communication protocol details of the IoT sensor network 27

Fig. 2.9 (A) LoRa Gateway LG01 (indoor unit LoRa), (B) Gateway LG02 28

Fig. 2.10 ThingSpeak account, channel and field structure and relationships..... 30

Fig. 2.11 ThingSpeak interface 1) Channel name 2) Channel identification number and access details with channel description 3) Channel status 4) Channel navigation 5) MATLAB visualization – last hour spectral reflectance signature of the 4th end node target 6) Field 1 of the ThingSpeak channel 1059365 which receive 4th end node target spectral response of the wavelength 410 nm..... 31

Fig. 2.12 Node 1 vegetation indices visualization 32

Fig. 2.13 A) Soil moisture telemetry data B) Battery voltage telemetry data 33

Fig. 2.14 IoT sensor node map and treatments including sensor node locations with node numbers and treatments with treatment numbers..... 35

Fig. 3.1 Total number of messages received by the ThingSpeack IoT platform via the LoRa gateway during June 2020 and September 2020 38

Fig. 3.2 Number of messages received vs number of messages transmitted by the senor nodes in a single hour..... 40

Fig. 3.3 Transmission efficiency vs number of messages sent per hour in the IoT network 41

Fig. 3.4 Proposed structural design for the sensor node to improve the current design (Design front view (left) and side view (right)) 42

Fig. 3.5 In-house AS7265x spectroscopy sensor calibration and testing results..... 44

Fig. 3.6 Location map of the PAR sensor and the up-looking sensor in the experimental site (a) weather station (b) up-looking sensor position. 45

Fig. 3.7 PAR Sensor output vs up-looking spectral sensor wavelength 560 nm output... 46

Fig. 3.8 AS7265x wavelength 560 nm output vs LI-190R quantum sensor PAR in 1- to -1 graph 47

Fig. 3.9 Diurnal NDVI moving average value in the node 48

Fig. 3.10 Seasonal peak NDVI with polynomial trendline (order 6) for Node 1 50

Fig. 3.11 NDVI variation comparison between treatments 51

Fig. 3.12 Manual NDVI Data and IoT Node NDVI Data Comparison Node 4 52

Fig. 3.13 Manual NDVI Data and IoT Node NDVI Data Comparison Node 1 52

Fig. 3.14 Manually collected NDVI vs IoT sensor node collected NDVI between August 7th and September 9 53

Fig. 3.15 Commercial meters for crop parameter measurement based on vegetation indices (a) Apogee chlorophyll concentration meter (b) SPAD 502 plus chlorophyll meter (c) atLEAF Chlorophyll meter (d) SRS - Spectral Reflectance Sensor 54

Fig. 3.16 AS72s65x response to incoming solar radiation in four differentnt timestamps within a day 56

LIST OF TABLES

Table 2.2 Sensor node power consumption breakdown 23

CHAPTER 1 GENERAL INTRODUCTION

1.1. Introduction to Internet of Things (IoT)

The internet defines as the global system for interconnected computer networks. It was born as a project to enable time-sharing computers by the United States Department of Defense in 1960 (Abbate, 2000). Network protocol development was the turning point of the history of the internet. Transmission Control Protocol (TCP) designed in 1973, allows to transmit data flow between two connected devices orderly and error freely. The Internet Protocol (IP) passes individual packets between devices (Cohen, 1979) and reduced the complexity of network hardware components such as repeaters, routers, switches, and hubs. The development of TCP and IP protocol was the first successful step of the internet. It is used on the internet to communicate between networks and devices with the timely introduction of other protocols.

According to (Statista, 2021), by the end of 2025, there will be 38.6 billion internet-connected devices. The internet provides several services such as email, file sharing, instant messaging services, world wide web (www), voice over IP, video on demand, etc. In 1985 Carnegie Mellon University Computer Science Department developed a method to check the availability and temperature of the Coke bottles in the department vending machine (Browning, 2018). This Coke machine is considered the first internet-connected device other than a computer.

The term “Internet of Things” was first specified by Kevin Ashton in 1999 (Ashton, 2009) to internet-connected devices with a unique identification number, which gathers data and shares with other devices in the network. Nowadays, computers, laptops,

smartphones, wearable devices, sensors, augmented reality glasses, or any device connected to the internet are considered IoT devices. IoT devices automate data collection and transmission. Applications of IoT are in the healthcare industry (Atzori, Iera, & Morabito, 2010), logistics (Jiang & Su, 2012), manufacturing industry, sports, communication, military and security, smart environment monitoring, building, and home automation, and agriculture (Katsoulas, Bartzanas, Kittas, & Tzounis, 2017).

An IoT device consists of mainly a power source, one or more sensors, a microcontroller or a processor, and communication hardware to collect, process, and transmit data acquired from the interacting environment. These devices can work without human involvement. The connectivity, networking, and communication protocols used with these internet-connected devices largely depend on the specific IoT applications employed (Rouse, 2020).

There are several advantages of IoT. They enable monitoring and actioning at a distance, consume very little energy, collect large amounts of data automatically, can be remotely controlled, and improve the quality of life and efficiency of processes. Furthermore, Ploennings et al. (2018) mentioned disadvantages of IoT such as IoT was technically complex, had issues with batteries, and data analytics at the edge did not develop well in 2018. Also, there were issues with data privacy (Ploennigs, Cohn, & Stanford-Clark, 2018).

1.2. IoT in Agriculture (Crop Production)

The advantages of IoT encouraged researchers to deploy IoT platforms in agriculture sensing and monitoring systems. According to Vasisht et al. (2017) data-

driven approaches increase yields while reducing losses and cutting down input costs. Therefore, IoT devices are ideal for data collection in agriculture, which is conventionally time-consuming and laborious, but the same authors also suggested that implementation of IoT sensor networks was discouraged due to high costs associated with the IoT systems, and senor networking issues due to distance and low bandwidth. Presently, the use of sensors that generate a high volume of data, such as RGB and multispectral cameras, are limited for IoT applications in agriculture.

When qualitatively assessing the recent researches about IoT applications in agriculture, we found that *in situ* soil water content measurement was the most assessed parameter ((Anghelof, Suciu, Craciunescu, & Marghescu, 2020); (Castellano, Deruyck, Martens, & Joseph, 2020); (Vasisht, et al., 2017); (Patil & Kale, 2016)). Several commercial companies have successfully developed and marketed IoT platforms for in situ soil, weather, and crop parameter measurement, monitoring, controlling, and decision-making. Some examples of these IoT systems and service providers are Adcon Telemetry GmbH, Agrosmart S.A., Arable Labs, Inc., Libelium Comunicaciones Distribuidas S.L., Davis Instruments Corporation, FarmX Inc., Iridium Communications, Inc., METER Group, Inc., Motorleaf Inc., Grownetics, Inc., Pycno, Ranch Systems, Inc., Spectrum Technologies, The Yield Pty Ltd, and Hortau, Inc.

Soil parameters such as; soil water content, soil conductivity, soil pH, soil temperature, environmental parameters such as; wet bulb and dry bulb temperature, relative humidity, wind speed, wind chill, atmospheric CO_2 level, atmospheric pressures, rainfall, solar radiation including (Photosynthetically Active Radiation (PAR) and UV),

crop parameters like; plant height, (stem, trunk, and fruit) diameter, leaf wetness and canopy temperature, become parameters that are commonly measured by commercial IoT platforms (Libelium, 2019). On the contrary, IoT applications for crop disease monitoring and pest monitoring/estimation are not common yet ((Rustia & Lin, 2017); (Spectrum Technologies, 2011)).

Even though there are state-of-the-art sensing systems available, the actual in-field IoT system implementation was limited because of the challenges for in-field sensor network development with long-distance high throughput data transmission capabilities. The high-power consumption for sensors and data transmission were the two other main demotivation factors.

1.3. Overview of Current IoT Systems

1.3.1. Existing Agricultural IoT System Architecture

Different IoT systems were proposed by researchers during the past few years. The system architecture depends on the type of communication technology available in the site, communication protocol, cost of hardware and software, land area implemented, crop type, open-field or indoor agriculture, sensors used, and whether the systems are research- or application-oriented.

According to Vasisht et al. (2017) the FarmBeats system consisted of two communication layers where the base stations in the farm were connected to the farmhouse using TV White spaces while they were connected to the field sensors (including cameras) and drones using Wi-Fi. TV White spaces are locally unused radio frequencies of commercially allocated frequencies for broadcast television channels. The

cloud service used was Microsoft Azure. To store data during internet failures, local storage at the farmhouse was used. Advantages of the system were no communication subscription fee for farmers other than the monthly internet cost, high data communication rate, availability of local data storage, and cloud data storage. Local data storage is a very important system component to eliminate data loss in real-world scenarios. Fig. 1.1 is a schematic diagram developed by Vasisht et al. (2017) including the relationships between key technologies, components, and features of the FarmBeats systems.

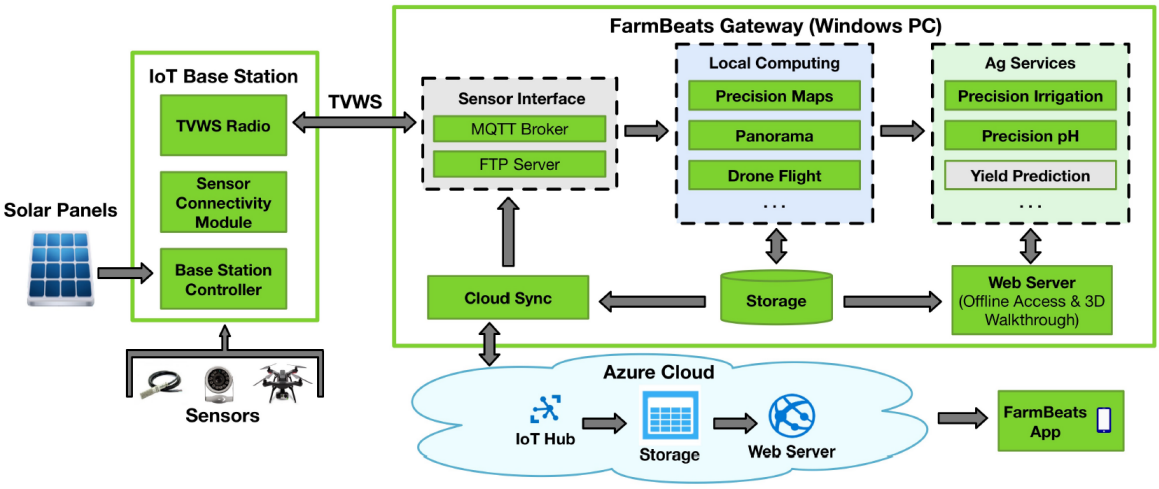

Fig. 1.1 Schematic diagram of the architecture of the FarmBeats IoT system **(Vasisht, et al., 2017)**

Disadvantages were the high subscription cost of the Azure IoT *Suite* and the small coverage area of Wi-Fi which led to more base stations in the field. Closed-loop control of irrigation based on sensor readings was also implemented in the system.

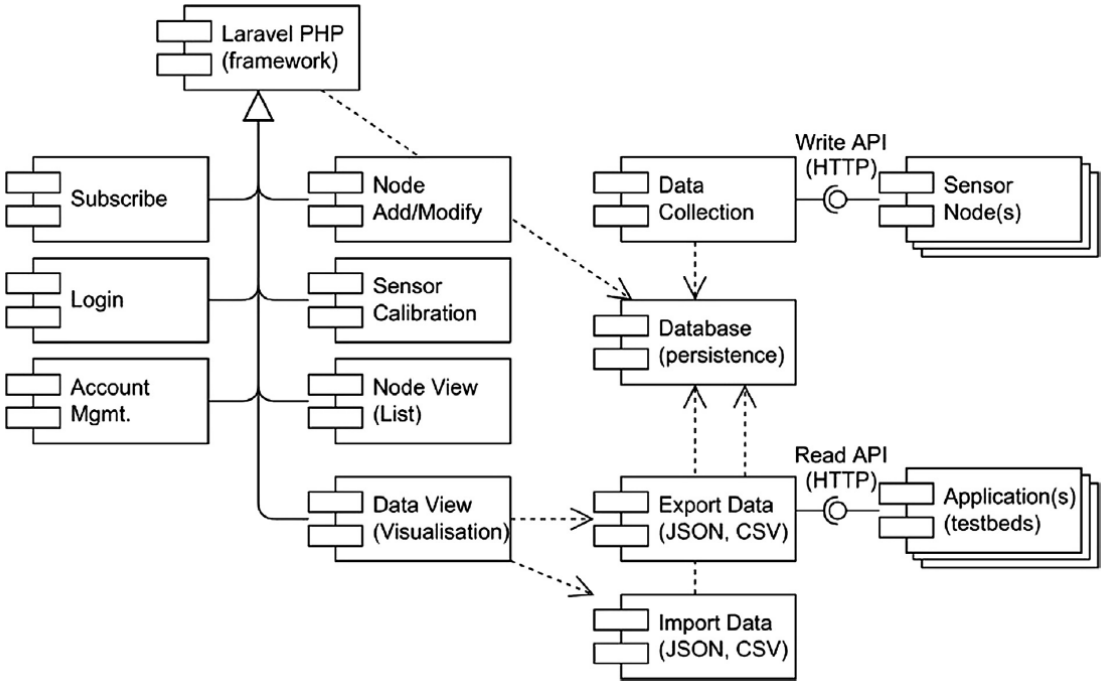

Fig. 1.2 System components of the Internet of Things (IoT) enabled platform for the research in precision agriculture and ecological monitoring from an implementation viewpoint **(Popovic, et al., 2017)**

IOT enabled platform architecture for the research in precision agriculture and ecological monitoring, was presented by Popovic et al. (2017) and it had similar architecture like the Farmbeats, but the sensor nodes designed by (Libelium, 2021) were integrated into the system. The (Popovic, et al., 2017) research paper was important because the authors discussed the IoT system design from the perspective of multiple, concurrent views. There the system architecture described from an implementation viewpoint as shown in Fig. 1.2 which discussed the ability to subscribe, log in, and manage the accounts. Next, the importance of expanding capabilities of the sensor network, software side sensor calibration (where the cloud data analyzing tools convert sensor readings to meaningful values or units based on calibration equation), data

collection, data visualization, and manual data downloading capabilities were discussed and demonstrated. This system was successfully implemented for precision agriculture, aquaculture, and ecological monitoring using Arduino and Raspberry Pi development boards. These capabilities are important for a farmer or a researcher during the IoT implementation stage.

Automated high throughput plant phenotyping platforms were also designed based on IoT principles. The IoT system architecture (framework) proposed by Fan, et al., (2020), can be considered as the most up-to-date framework for an agricultural IoT system for plant phenotyping. The author discussed the integration of industrial control computers with sensors (examples: - lidar, multispectral camera, Kinect, fluorometer, and thermal camera), edge computing gateways (ex: - environmental sensors), embedded terminals (examples: - human-machine interface displays and 3D monitors), point to point LoRa data terminal clusters, monitor cluster switches, and network bridges to cloud computing system through an internet-connected router or switching system. This integrated design will be commercially viable soon with the introduction of low-cost sensors.

1.4 IoT System Layers

Typical IoT system architecture is based on three layers; namely, the perception layer (sensors and actuators), the network layer (data transfer), and the application layer (data storage and manipulation) (Tzounis, Katsoulas, Bartzanas, & Kittas, 2017). Additional two layers were also important to complete an IoT system and they were the

middleware layer which is in between the network layer and the business layer, after the application layer (Vashi, Ram, Modi, Verma, & Prakash, 2017).

1.4.1 Perception Layer

The perception layer consists of sensors and actuators. This layer is responsible for collecting and processing transmittable data. Fig. 1.3 defined the general architecture of perception layer devices. The perception layer contributes to the spatial and temporal resolution of data collection in agriculture.

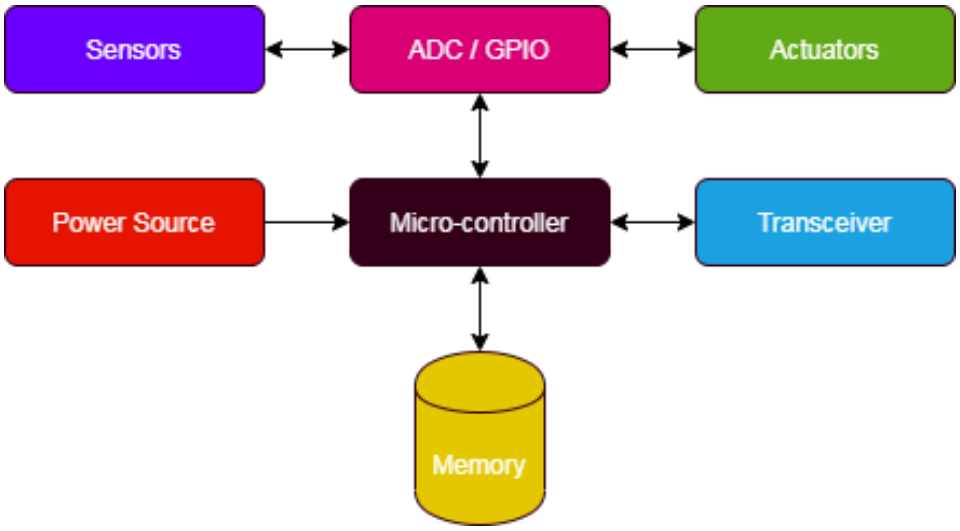

Fig. 1.3 Perception layer device architecture

IoT perception layer devices are unique compared to traditional sensors because they have a unique identification number to represent it on the internet (AliKhattak, AliShah, Khan, Ali, & Imran, 2019). As an example, Wi-Fi-connected sensor nodes represent their uniqueness through IP address while LPWAN perception layer devices have a unique identification number.

1.4.2 Network Layer

The network layer is used to transmit data collected by sensors through an existing communication network from the perception layer to the application layer, whereas decisions or control signals are sent from the application layer to actuators in the perception layer (Leo, Battisti, Carli, & Neri, 2014). Bidirectional communication is not necessary for systems to monitor only. The network layer has two main components: Communication Technology (CT) and the Communication Protocols.

Wired CT refers to the data communication technologies through physically connected fiber optics or metal wires, such as USB, RS-232, SPI, and RJ45, which usually have less signal attenuation but the higher cost in implementation and maintenance.

Wireless CT is less costly but suffers high signal attenuation and has complex protocols. Wi-Fi, ZigBee, 6LowPAN, Z-wave, 2G/3G/4G/5G, LPWAN, LTE-M cellular technologies (Gungor, et al., 2011) underground wireless communication are examples of Wireless CT (Vuran, Salam, Wong, & Irmak, 2018).

Multiple standardized agencies such as the International Telecommunication Union (ITU), the Institute of Electrical and Electronics Engineers (IEEE), introduced different communication protocols for the network layer of IoT (Ejaz & Anpalagan, 2018). The Low Power Wide Area Network (LPWAN) protocols have become very popular due to their advantages such as low power consumption, long-range communication capability, and low cost.

SigFox, Cellular Protocols such as 3G/4G/LTE, NB-IoT, LoRa, Weightless are some common communication protocols suitable for IoT (Al-Sarawi, Anbar, Alieyan, & Alzubaidi, 2017). Depending on the applications the network layer protocols must be selected. Common design factors are the distance between nodes, bandwidth, cost, and infrastructures required.

1.4.3 Middleware layer

According to (Ngu, Gutierrez, Metsis, Nepal, & Sheng, 2017) there are three types of IoT middleware. The first type is service-based which has a service-oriented architecture (SOA). It allows developers and users to deploy multiple IoT devices as services and LinkSmart is an example (LinkSmart, 2021). The second type is a cloud-based solution that limits the users on the type and the number of IoT devices that they can deploy but allows users to connect, collect, and interpret the collected data easily since certain use cases can be determined and programmed without any understanding about the sensor data. Xively is an example of cloud-based middleware architecture (Sinha, Pujitha, & Alex, 2015). The actor-based framework that emphasizes the open, plug-and-play IoT architecture is the third type. A variety of IoT devices can be exposed as reusable actors and distributed in the network.

1.4.4 Application Layer

Data formatting and presenting are the main role of the application layer. The application layer has protocols that handle the data communication between the nodes, gateways, and cloud data storage, as well as between cloud data storage and end-user applications. Basic actions handled by these protocols are data request, response, publish

and subscribe. CoAP, MQTT, XMPP, REST, AMQP, and Web socket are commonly used application layer protocols (Karagiannis, Chatzimisios, Vazquez-Gallego, & Alonso-Zarate, 2015).

1.4.5 Business Layer

Sustainable management of the entire IoT system, including applications, business and profit models, and users' privacy are the main purposes of the business layer (Navani, Jain, & Nehra, 2017). The scope of the business layer is somewhat less relevant for this thesis.

1.5 Monitoring Crop Canopy Vegetation (Pigment) Indices

Monitoring crop canopy pigment indices is important to understand crop status. Normalized difference vegetation index (NDVI) was used to estimate Leaf Area Index (LAI) (Nguy-Robertson, et al., 2012), photosynthesis and biomass accumulation (i.e., carbon uptake) (Monteith, 1977), canopy productivity (Ryu, et al., 2010), while Photochemical Reflectance Index (PRI) was used to estimate Gross Primary Productivity (GPP) of ecosystems, Light Use Efficiency (LUE), and chlorophyll to the carotenoid content ratio (Porcar-Castell, et al., 2012).

(Gamon, Kovalchuck, Wong, Harris, & Garrity, 2015) stated that automated, ground based low-cost NDVI, and PRI sensors offer new opportunities for monitoring crop canopy pigment indices. They are important because they could accurately and reliably obtain the diurnal and seasonal change of these variables compared to the conventional Vegetation Index (VI) monitoring techniques from aerial, airborne, and satellite remote sensing. For the authors' knowledge METER's Spectral Reflectance

Sensor (SRS) is the commonly used in-field spectral reflectance sensor in the market that measures NDVI and PRI (METER Group, SRS Spectral Reflectance Sensor Operator's Manual, 2020) but no other VIs.

With the introduction of several other vegetation indices with different capabilities such as Dynamic Range Vegetation Index (WDRVI), Chlorophyll Absorption Ratio Index (CARI), Visible-Band Difference Vegetation Index (VDVI), etc., it is a timely important task, to develop a low-cost sensor system with multispectral reflectance logging capability, for in-field crop monitoring. This kind of product will be useful for crop researchers and farmers. Furthermore, crop canopy vegetation index monitoring is an important input for precision agriculture.

1.6 Conclusion

IoT is an important and rapidly evolving technology that has many applications in different fields including agriculture. During the last three decades, VI monitoring employing satellite, handheld sensors, ground-based systems, mobile platforms, and Unmanned Aerial Vehicles became popular for agricultural and forestry applications. Due to less awareness and the high cost of field VI monitoring systems which include multispectral sensors, a significantly fewer amount of farmers uses these systems. Moreover, real-time telemetry data is important for farmers to make decisions. When considering most of the farming lands they lack communications infrastructures to implement state of the art sensing systems.

LPWAN network technology such as LoRa and reliable low bandwidth data communications protocols such as MQTT can be used to develop an IoT-enabled sensor

network with a suitable multispectral sensor. Such techniques and tools will be important for farmers as well as researchers to monitor crops easily in real-time, analyze data, and make timely management decisions.

Based on the literature review and past in-field data collection experience the author had, two objectives were set up initially.

The first objective was to develop an IoT-enabled, long-range communicable, in-field wireless sensor network with the capabilities of real-time VI and soil water content sensing.

The second objective was to do a performance evaluation of the developed system on the aspects of sensing and communication.

VI sensing accuracy, precision, range, and resolutions were sensor performance evaluation parameters while long-range communication distance, data transmission rate, and data loss are communication technique performance evaluation parameters.

The outcomes of this research would be important to identify the advantages and limitations of a large-scale ground sensor network for infield crop monitoring.

This research was motivated by the fact that the outcome of this research will be beneficial for farmers to improve crop productivity. Recent advancements of low-power-consuming communication technology and spectral sensor development became other motivations to start the research.

CHAPTER 2 DEVELOPMENT AND FIELD TESTING OF THE NOVEL IOT SENSOR NETWORK FOR INTEGRATED SOIL, CROP, AND ENVIRONMENT SENSING

2.1 Introduction

Chapter two explains the overall solution developed as the answer to the research question identified in chapter one followed by the experimental design. Sensor selection, signal processing unit selection, communication module selection, sensor calibration steps, power and energy supply selection, data storage, data transmission, cloud data storage system selection, cloud data processing, and system field testing will be discussed.

2.2 Hardware Selection and System Design

We planned to include three sensors: a multispectral sensor, an RGB camera, and a soil water content sensor to achieve the first objective and to demonstrate it. The selected long range communication technology is LoRaWAN.

RGB cameras are sensitive to visible spectrum wavelengths with peaks of blue (460 nm), green (540 nm), and red (620 nm) wavelengths. However, incorporation of the RGB sensor was done, not to demonstrate the ability of VI derivation but to demonstrate the power of edge computing image processing techniques during the research. However, within this thesis, this outcome will not discuss in detail.

2.2.1 Spectroscopy Sensor Selection

The spark fun triad spectroscopy or the spectrophotometer sensor board in Fig. 2.1 consists of three sensors; AS72651 with master capability, AS72652, and AS72653, which were designed and manufactured by Austria Mikro Systeme. Each sensor can measure the intensity of six individual electromagnetic wavelengths as given in Fig. 2.2 AS7265x photodiode arrays, in counts/µW/cm^2 units with ±12% accuracy. The temperature sensor in each sensor helps to correct the effect of temperature from the spectral output (AG, 2018).

Fig. 2.1 Spark fun triad spectroscopy sensor board

The full width at half maximum of the spectroscopy sensor is typically 20 nm and the wavelength accuracy are ±10 nm. The average field of view of the sensor is ±20.5 degrees. The spectroscopy sensors are pre-calibrated with a specific light source according to the manufacturer details.

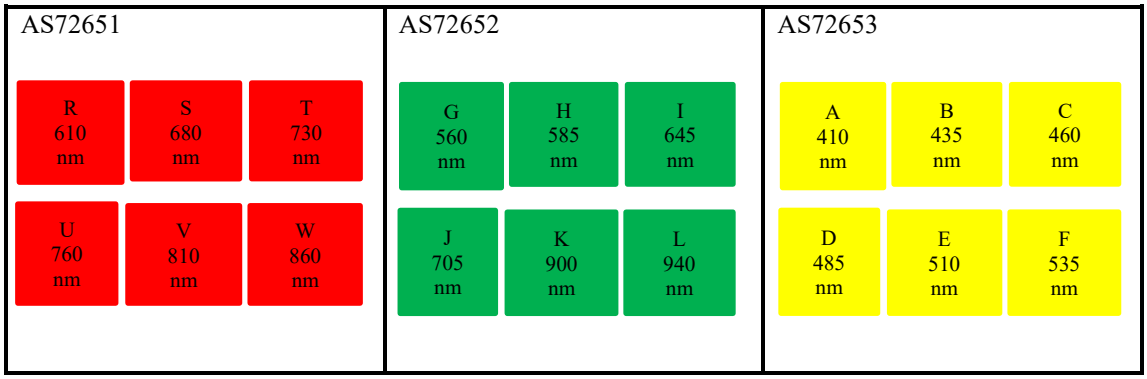

Fig. 2.2 AS7265x photodiode arrays sensitive wavelengths with reference letter

Low power consumption, low operating voltage (3.3 V), compact design, built-in aperture, not demanding additional signal conditioning, I2C (Inter-Integrated Circuit), and serial communication capability are the main reasons to embed the Spark Fun Triad Spectroscopy to the IoT nodes at the design stage.

Down-looking spectroscopy sensors in the IoT nodes capture and measure the electromagnetic energy reflected from the vegetation and soil. OEM recommendation is that the AS7265x sensor is not suitable for high-intensity UV (ultraviolet) environments without UV filters. Therefore, the up-looking sensor node was covered with a Lambertian diffuser to measure the incoming solar radiation. The down-looking spectroscopy sensors were not covered with Lambertian diffusers assuming the reflected light from soil and vegetation were diffuse.

The down-looking spectroscopy sensor on the sensor node could be moved vertically from the ground level up to 3.3 m along the supporting pole as shown in Fig. 2.3. The Ground Instantaneous Field of View (GIFOV) of the spectral sensor can be calculated using equation 2.1.

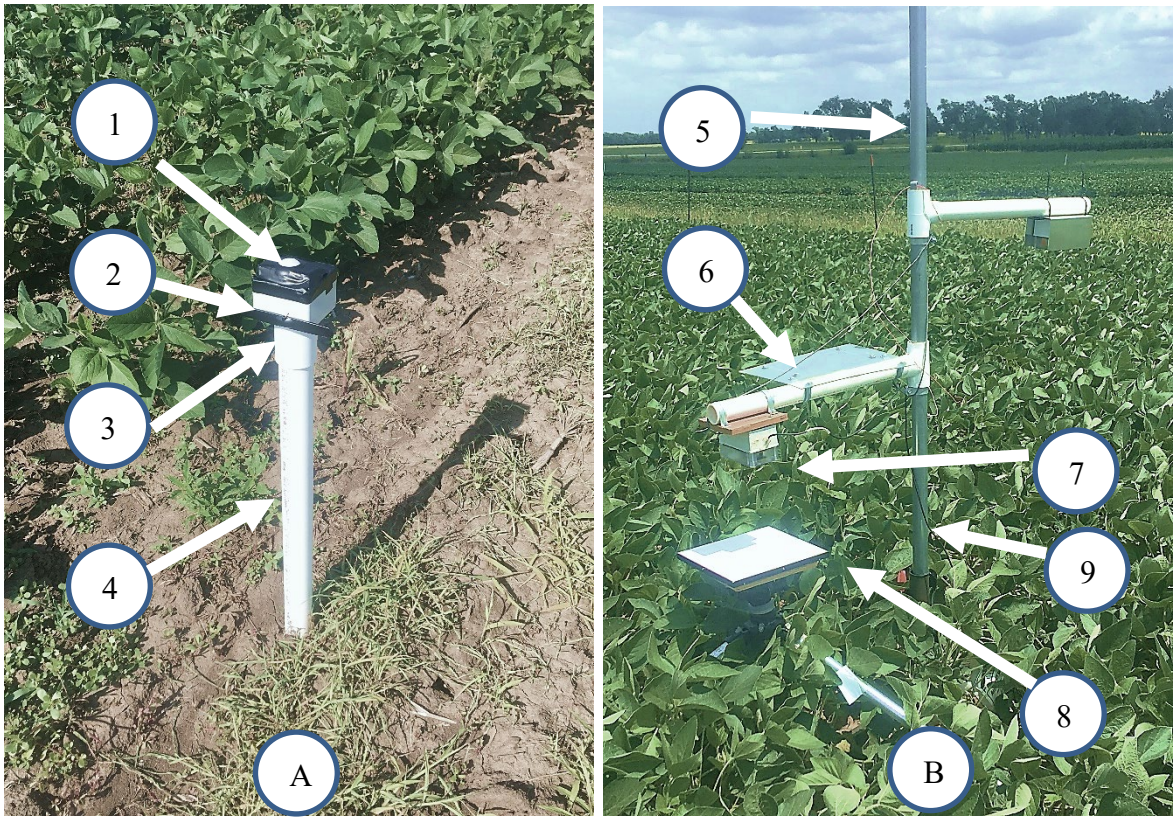

Fig. 2.3 The up-looking senor node (A) and the down-looking sensor node (B) of the IoT sensor network. The components of the sensor nodes included (1) Lambertian diffuser (2) Circuit enclosure of the sensor node (3) LoRa Antenna (4) Supporting pole (5) Supporting pole for down looking sensor (6) Solar panel (7) Down looking sensor node (8) Sensor calibration Lambertian surface (9) Soil water content sensor cable

$$\text{GIFOV} = 2H \times \tan\beta \qquad \text{(Eq. 2.1)}$$

where H is the vertical height from the target (ex:- soil, canopy) to the sensor and β is the Instantaneous Field of View (IFOV) of the spectroscopy sensor as shown in Fig. 2.4 Average Field of View of the Spectroscopy Sensor

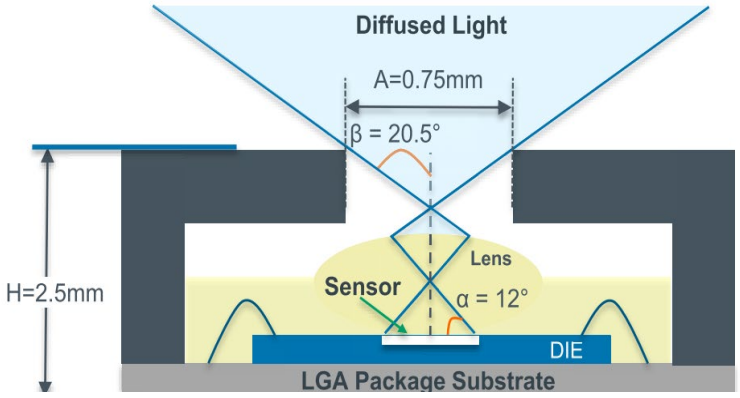

Fig. 2.4 Average Field of View of the Spectroscopy Sensor **(AG, 2018)**

2.2.2 Spectroscopy Sensors Calibration

Spectroscopy sensors calibration was performed using a calibration panel placed under the node as shown in Fig. 2.3. The calibration panel was placed to cover the sensor's GIFOV (that is, GIFOV < the shorter side of the calibration panel). One critical issue faced during the calibration was that the 410 nm channel exceeded the maximum value the channel could hold when the calibration surface was exposed to direct sunlight. To avoid this issue the calibration readings were recorded under a cloudy environment when the diffuse sunlight reaches the calibration panel. This leads to the peak reflectance above 100% for some wavelengths.

$$CF_X = 100\% \times \frac{UL_X}{DL_X} \qquad \text{(Eq. 2. 2)}$$

where CF_X is the wavelength-specific correction factor; UL_X is the wavelength-specific downwelling irradiance intensity; DL_X is the wavelength-specific radiance intensity from the calibration panel, and x is the 18 channels of the sensor.

2.2.3 Reflectance Calculation

Percent reflectance for each measured wavelength was calculated real-time using the ThingSpeak IoT Platform with MATLAB Analytics following the equation 2.3.

$$Reflectance\ \% = \frac{DL_x}{UL_x} \times CF_X \quad \text{(Eq. 2.3)}$$

2.2.4 Vegetation Indices Calculation

NDVI and Green Chlorophyll Index (GCI) were calculated in real-time. The value of NDVI ranges from -1 to 1. GCI is used to estimate leaf chlorophyll content across a wide range of plant species. A NIR wavelength at 800 nm and a red wavelength at 680 nm from the spectral sensor were used to calculate NDVI. NDVI can be calculated by equation 2.4. Refer to the Appendix section 2 for the MATLAB NDVI calculation program.

$$NDVI = \frac{(\rho_{NIR} - \rho_{RED})}{(\rho_{NIR} + \rho_{RED})} \quad \text{(Eq. 2.4)}$$

In equation 2.4 ρ_{NIR} is the reflectance at near-infrared wavelength and ρ_{RED} is the reflectance in visible red wavelength.

$$GCI = \frac{(\rho_{NIR})}{(\rho_{Green})} - 1 \quad \text{(Eq. 2.5)}$$

Having broad NIR and green wavelengths provides a better prediction of chlorophyll content while allowing for more sensitivity and a higher signal-to-noise ratio (A. Gitelson, Gritz, & N. Merzlyak, 2002).

2.2.5 RGB Camera Selection

The five Mega Pixel CMOS (Complementary Metal Oxide Semiconductor) imaging sensor (OV5642, Omni Vision Technologies, Inc., California, USA) had been

used for different imaging applications for a long time and proved successful for external environmental applications (Jianjun, Deqin, & Shunbin, 2014); (Rojas, Palomino, & Chávez, 2017). Therefore, we used the OV5642 image sensor which is shown in Fig. 2.5 with ArduCam Camera Shield (www.arducam.com) to collect images. The camera shield was fixed to the ArduCAM ESP8266 UNO board to capture RGB images under the sensor node. Captured images were saved in the SD card on the ArduCAM ESP8266 UNO board to conduct image processing on the node and to transmit the processed image information such as leaf area coverage and the presence of pest/diseases.

Fig. 2.5 ArduCam camera shield RGB camera with a five Mega Pixel CMOS (Complementary Metal Oxide Semiconductor) imaging sensor (OV5642, Omni Vision Technologies, Inc., California, USA)

One important feature of this camera shield is that it controls the camera through the commands received through I2C communication protocol while imaged data were transmitted via SPI (Serial Peripheral Interface), which is the de facto standard for short-range communication in embedded systems.

2.2.6. Soil Moisture Sensor

The TEROS 10 Soil Moisture sensor (METER Group, Inc. USA) was used to measure volumetric water content (VWC). It was buried 15 cm underground close to each sensor node. The operating voltage of this sensor was in the range of 3.0 to 15.0 VDC and the output was an analog signal in the range of 1,000–2,500 mV.

This sensor uses an electromagnetic field to measure the apparent dielectric permittivity (Ɛa) of the surrounding medium, in this case, soil. The sensor supplies a 70-MHz oscillating wave to the sensor needles, which charge according to the dielectric property of the material. The charge time is proportional to substrate dielectric property (primarily determined by VWC for soil). The TEROS 10 microprocessor measures the charge time and outputs a raw value based on the substrate Ɛa. The raw sensor output is converted to VWC by Eq. 6.

$$\Theta\ (\ m^3/\ m^3\ ) =\ 4.824 \times 10^{-10} \times mV^3 - 2.278 \times 10^{-6} \times mV^2\ + 3.898 \times 10^{-3}\ \times mV - 2.154 \quad \text{(Eq. 6)}$$

where m3/ m3 is the unit of volumetric water content, mV is TEROS 10 sensor output voltage in millivolts. VWC of the soil changed from 0 to 0.64 m^3/ m^3 where 0 is no water at all and 0.64 is saturated. Sometimes when the sensor was hanging on air its' output becomes negative.

2.2.7 Integrated Sensor Node Design Analysis

The computation unit of the IoT sensor node was Atmel SAMD21 (Microchip Technology Inc., Arizona, USA). This low-power microcontroller was embedded in the

MKR WAN 1300 microcontroller development board (Arduino LLC) with a CMWX1ZZABZ Lo-Ra module (Murata Investment Co., Ltd., China).

MKR WAN 1300 accepted 5VDC via a USB input from the SPM (Solar Power Manager) as shown in Fig. 2.6. The SPM (DFRobot, Shanghai, China) had a 5 V constant USB power output and 5 V controllable outputs using GPIO (General Purpose Input Output) pins. MKR WAN 1300 was powered through the USB power supply while the ArduCAM module and the sensors were powered through the controllable voltage outputs (Fig. 2.6 link number 2, 3, and 5). Therefore, the MKR WAN 1300 can turn on and off the sensors (including the ArduCAM ESP8266 UNO board) if needed to save the power using the GPIO pins (Fig. 2.6 link number 7). SPM has an analog battery voltage output (Fig. 2.6 link 7). It allows the MKR WAN 1300 to determine the battery capacity and go to sleep mode while turning off the sensors.

The maximum power consumption of the IoT sensor node was 5.458 W and the minimum power consumption was 0.14765 W (refer to Table 2.2 Sensor node power consumption breakdown). Assumptions made during this calculation were that under maximum power consumption all the power consumers in the node would consume the highest rated power and under minimum power consumption MKR WAN 1300 would stay in the sleep mode and RTC would stay idle.

It is important to note that the peak power demand of sensors occurs when the sensors take readings actively. Typical sensor reading time does not exceed 1 second. MKR WAN 1300 was programmed not to turn on all the sensors at once and there was a

10-second delay between each sensor measurement. The order of activation is the spectroscopy sensor first, the soil moisture sensor second, and the imaging sensor last.

Table 2.1 Sensor node power consumption breakdown

Category	Module Name	Voltage V	Current mA	Power Consumption / Handling Maximum (w)	Power Consumption Handling Minimum (w)
Power Consumers	Spectroscopy Sensor	3.3	13 min /113 max	0.3729	0.0429 (Idle) 0 (Off)
	Soil Moisture Sensor	3.3	12	0.0396	0 (Off)
	ArduCAM Shield	5	20 min / 390 max	1.95	0.1 (Idle) 0 (Off)
	ArduCAM ESP8266 UNO board	5	370 Typical	1.85	0 (Off)
	MKR WAN 1300	5	29 min / 74 max	0.1448	0.00373 (Sleep)
	RTC (Real Time Clock)	3.3	.11 min/ .2 max	0.00066	0.00036
	SD card interface	3.3	0.2 min / 333 max	1.1	0.00066 (Idle)
Power Distributor	SPM (out)	5	1500 max *	7.5	0 (Off)
Power Sources	(LiPo) Lithium-ion Polymer Battery	3.7	760 **	2.812	0 (Fully Discharged)
	Solar Panel	5	900	4.5	0 (Off)

This is the maximum allowable power supply by the SPM *
This is the recommended standard discharge current rate under 0.2C discharge rate**

The frequency of data collection for the down-looking sensor node was two readings per hour in the daytime and no readings during the nighttime. Time was measured by the Real-Time Clock (RTC) (Maxim Integrated Products, Inc., California, USA).

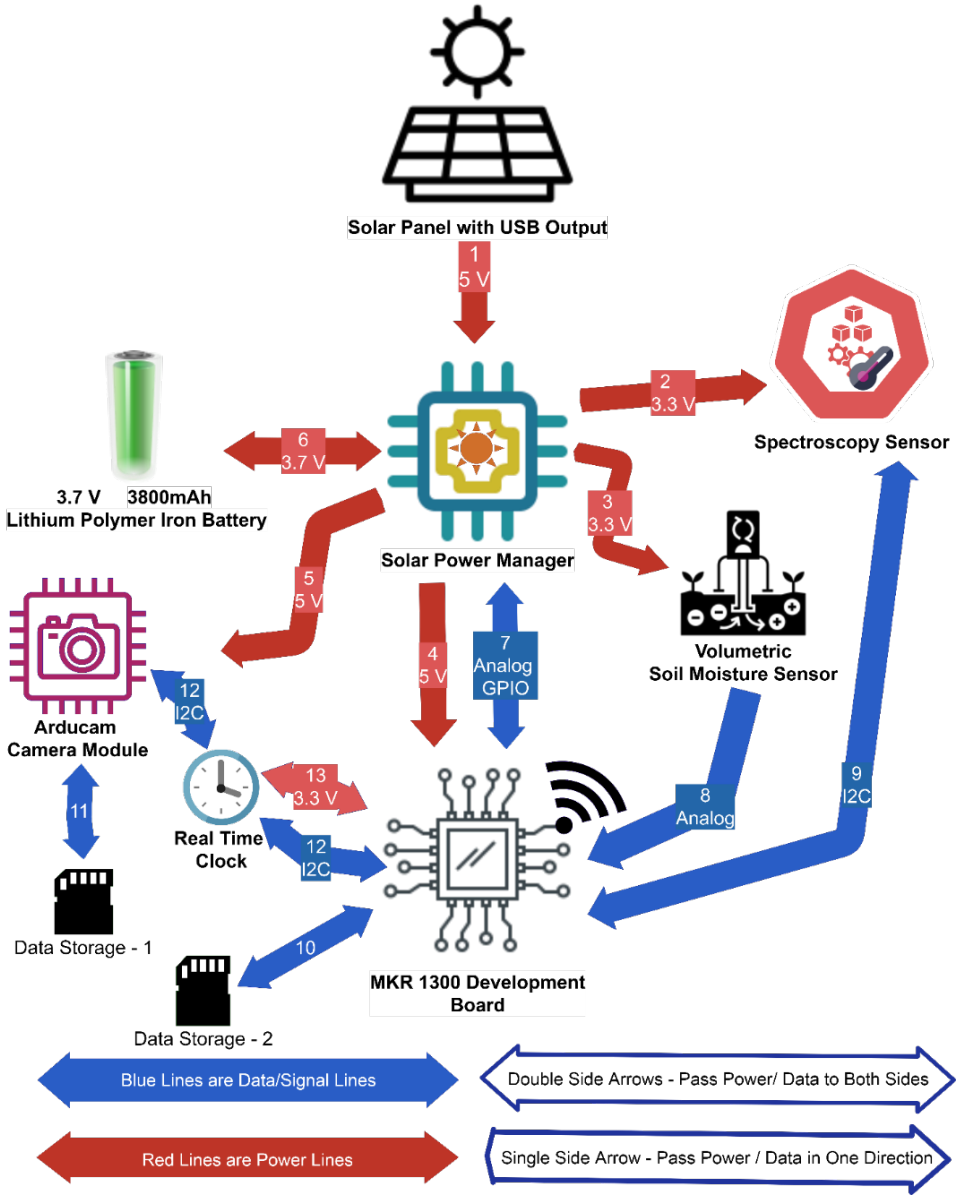

Fig. 2.6 IoT sensor node power supply lines, data flow lines, and control signal lines visualization

The up-looking sensor node only had a single spectroscopy sensor; therefore, it had enough power to collect spectral data at a higher frequency (3 readings per hour). Images from the RGB camera were saved in the SD card (Fig. 2.6 link 11) in the JPEG

(Joint Photographic Experts Group) format after a 1-second delay following image capturing.

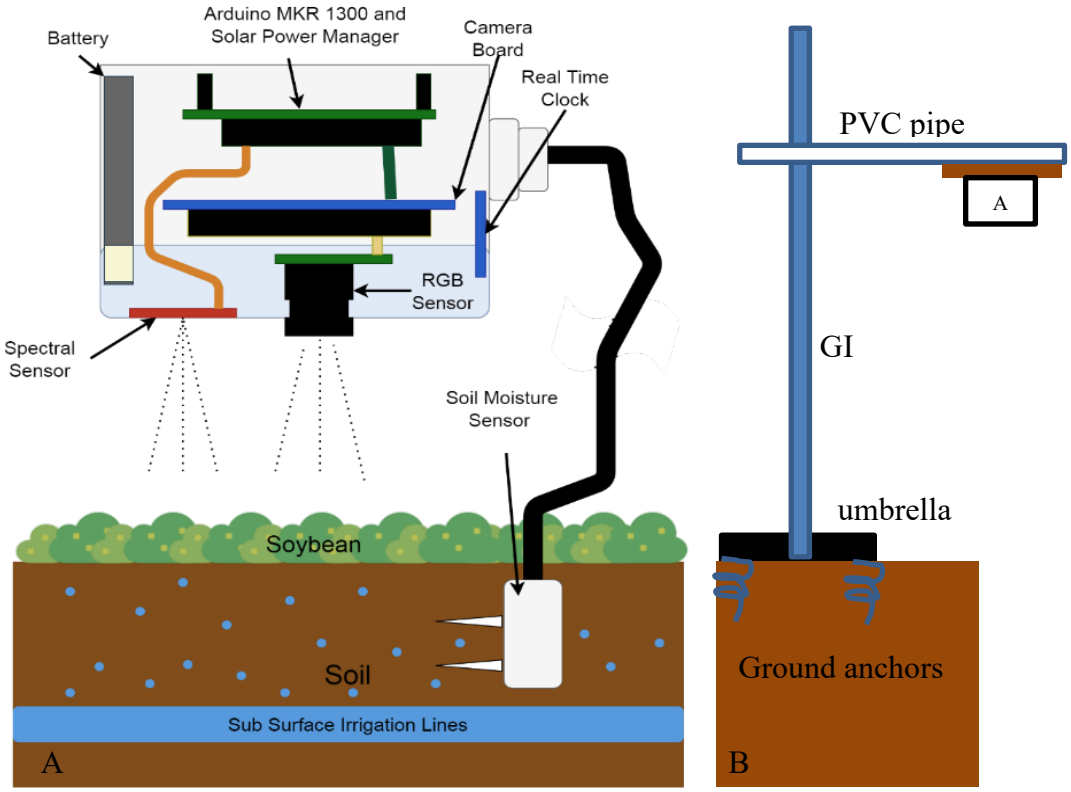

Fig. 2.7 Schematic diagram of the IoT sensor node structure with main components A) detailed view of node electronics, B) detailed structural arrangements

At a later stage, we discovered that the whole set of data can be sent via a single

The total number of data points collected in one reading cycle was 22 (18 spectral responses, 2 temperature values, 1 soil moisture value, and 1 battery voltage). Temperature values were the average spectroscopy sensor temperature and main IC temperature of the spectroscopy sensor. Data points were first saved in the SD card (Fig. 2.7 link 10). Data points were transmitted in 22 instances. The time interval between two data transmission instances was initially 1 minute. The up-looking sensor node transmitted 20 data points with a 20-second interval between 2 consecutive data points

and the reading frequency per minute is around 10. These timing intervals were kept constant during the experiment.

2.2.8 Sensor Node Physical Arrangement

All the electronics were enclosed in an IP65 electrical enclosure (size:158 x90 x60 mm), made with ABS plastic (Fig. 2.7 A). The enclosure was fixed to a lumber plate to keep the sensors target to the ground. The lumber plate was attached to a GI pole with a diameter of 3.81 cm using PVC pipe fittings. The GI pipe was attached to a steel outdoor patio umbrella base anchored to the soil with 20.3 cm anchors (Fig. 2.7).

2.3 Wireless Sensor Network Design

Arduino MKR WAN 1300 as the end node (Sensor Node 1 to 11 in Fig. 2.8) and OLG02 Outdoor Dual Channels LoRa IoT Gateway (Dragino Technology Co., LTD., China) (LoRa Gateway in Fig. 2.8) were the two main LoRa devices used to set up the physical network layer.

There were two types of LoRa gateways available, OLG02 for outdoors and the LG02 for indoors. LG01 is for single-channel communication while LG02 is for dual-channel communication. The dual channel allows receiving data via two channels. These devices support 200 to 300 end nodes according to the manufacturer (Dragino Technology Co., LG01 LoRa Gateway User Manual, 2018).

The OLG02 was selected as the LoRa gateway because it was an open-source dual channels LoRa Gateway. It was possible to connect it to the internet through different interfaces such as LAN, Wi-Fi, 3G, or 4G. Wi-Fi interface was selected during this development because Wi-Fi was available at the test site (the Farmhouse Fig. 2.8).

Both end nodes and the gateway were able to communicate both ways. However, in this research, this duplex capability was not tested. The end node directly sent the telemetry data to the gateway. The gateway then forwarded the data to the IoT server through the internet. Out of the available channels in the LoRa gateway, a single channel was used with 915 MHz frequency for setting up the simplex communication between the nodes and the gateway.

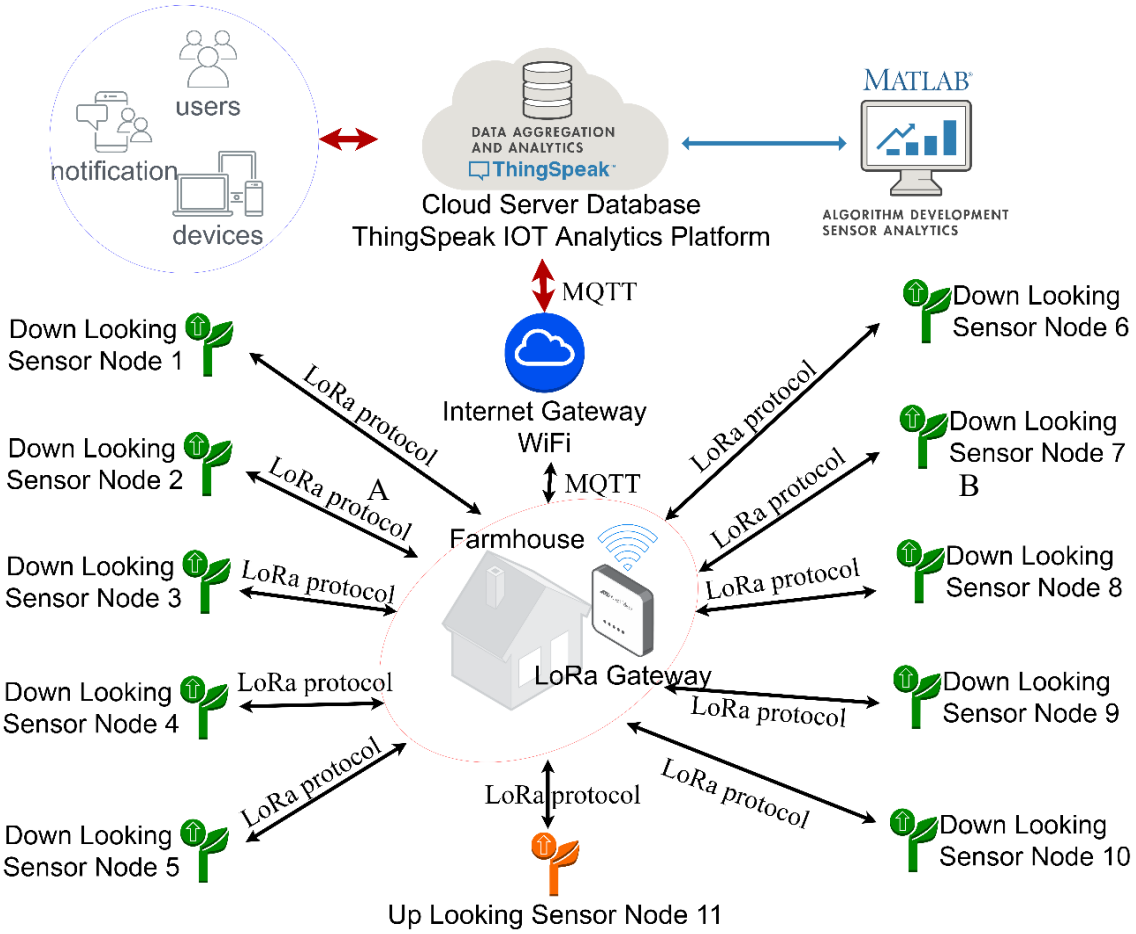

Fig. 2.8 Architecture and communication protocol details of the IoT sensor network

OLG02 had three built-in modes to set up wireless networks: LoRa repeater mode, Message Queuing Telemetry Transport (MQTT) mode, and the TCP/IP Server

mode. MQTT mode was selected to design the network and it allowed to store the sensor data (although a limited storage space) in the OLG02 until they are passed to the cloud storage.

Fig. 2.9 (A) LoRa Gateway LG01 (indoor unit), (B) Gateway LG02 (outdoor unit)

End nodes communicated with the LoRa gateway through the LoRa protocol while the LoRa gateway sent sensor data through the internet following MQTT protocol to the IoT server (Fig. 2.8). Programming of the nodes to send data to the LoRa gateway was developed based on the examples given in the LG02/OLG02 LoRa Gateway User Manual (Dragino Technology Co., LG02 LoRa Gateway User Manual, 2019).

2.3.1 The Quality of Service (QoS)

The quality-of-service level is an understanding between the transmitter of a message and the receiver of a message that defines the assurance of delivery for a certain message according to (Toldinas, Lozinskis, Baranauskas, & Dobrovolskis, 2019).

Further, there are 3 QoS levels in MQTT: QoS level 0 level ensures a best-effort delivery and there is no guarantee of delivery. The receiver does not acknowledge receipt of the message and the message is not stored and re-transmitted by the transmitter. QoS level 1 ensures that a message is delivered at least one time to the receiver. QoS level 2 is the highest level of service in MQTT. This level guarantees that each message is received only once by the intended recipients. In our sensor network, QoS 0 level was used because it was the accepted QoS level in the IoT analytics platform.

2.4 IoT Analytics Platform

The selection of a cloud service provider was a crucial task for the IoT sensor network since the success of data analytics and visualization completely depends on this matter. Amazon Web Services, Google Cloud Platform, and Microsoft Azure were on the selection list but dropped due to complexity and the high cost. ThingSpeak IoT analytics platform service provided by The MathWorks, Inc. was selected due to its capability of live sensor data collection, visualization, and analysis at an affordable cost in the cloud. Furthermore, ThingSpeak can read its data if anyone has its Channel ID and the access keys. This capability was successfully used during the research by installing the Thingview app, which was developed by Cinetica tech SRL on the researchers' mobile phone. Thingview app allowed the user to visualize data in its app interface. It was easier to use the app than loading the ThingSpeak website during node repairing and troubleshooting in the field.

2.4.1 ThingSpeak Setup, Features, and Specifications

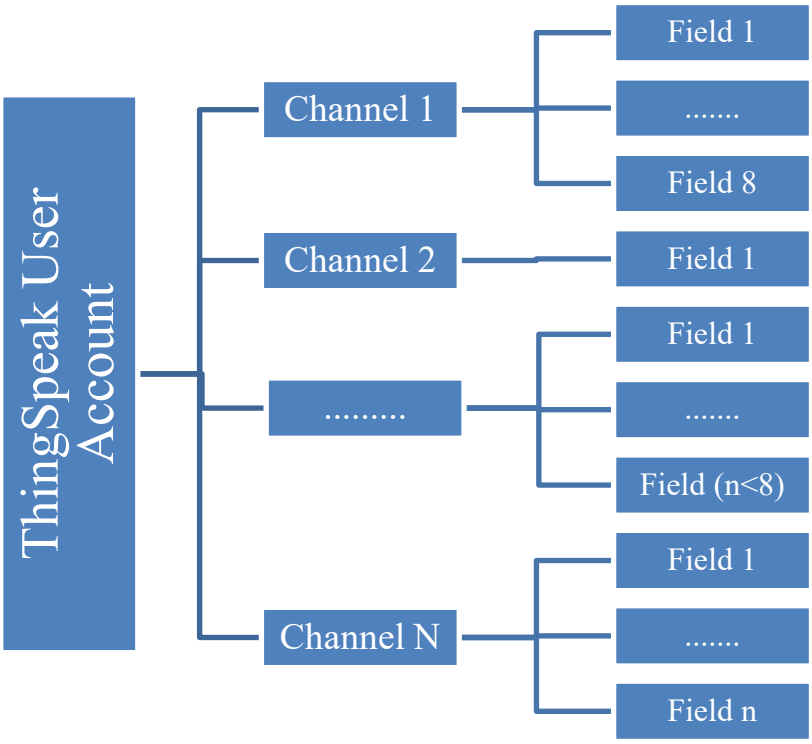

Fig. 2.10 ThingSpeak account, channel and field structure and relationships

ThingSpeak used fields to publish sensor telemetry data wherein a single ThingSpeak channel there are 8 fields as shown in Fig. 2.10. Each field has a field number, and a channel has a unique channel ID, a write API (application programming interface) key, and a read API key. A channel allows a maximum of 8 fields with telemetry data.

Additional fields such as metadata can be entered according to user requirement. Some metadata fields are introduction about channel data, including JSON, XML, or CSV data, tags, enter keywords that identify the channel, Uniform Resource Locator

(URL): where to add a website that contains information about the ThingSpeak channel and channel location: where to enable entering channel location data.

To write data to a field in a channel in ThingSpeak, the sensor node needs to send, API key, device ID, channel ID, and field id. Therefore, each node was installed with a program that included a unique 3 different channel IDs and relevant API keys.

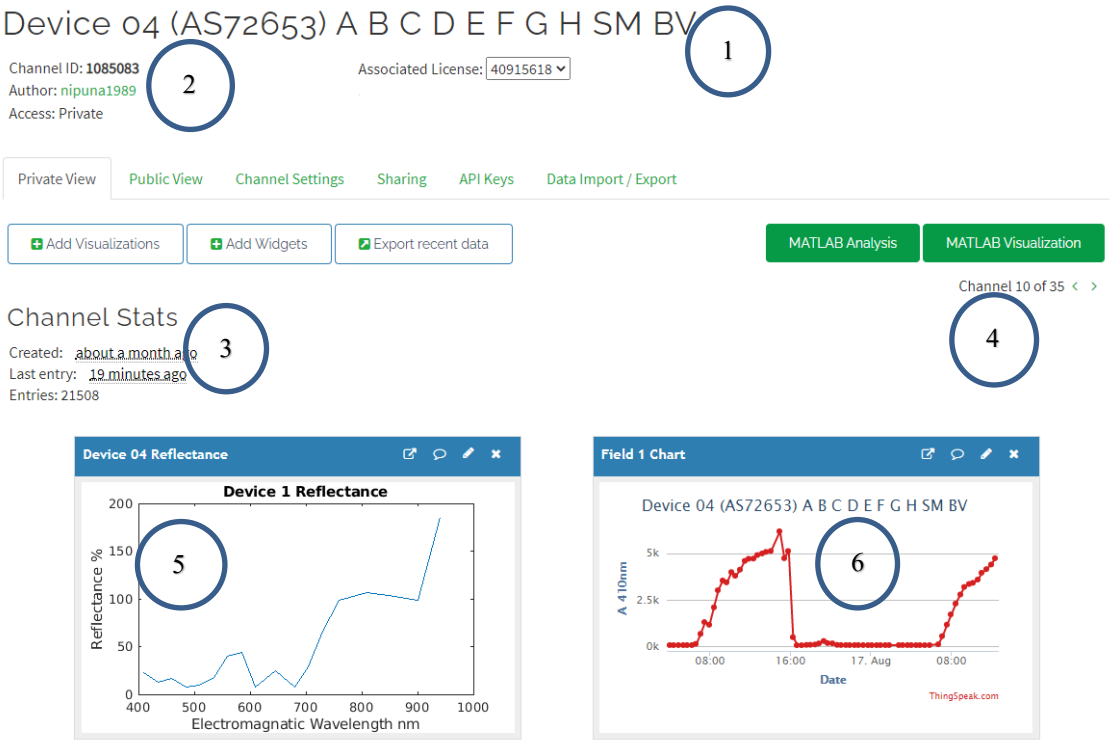

Fig. 2.11 ThingSpeak interface 1) Channel name 2) Channel identification number and access details with channel description 3) Channel status 4) Channel navigation 5) MATLAB visualization – last hour spectral reflectance signature of the 4th end node target 6) Field 1 of the ThingSpeak channel 1059365 which receive 4th end node target

The LoRa gateway identified the CHANNEL ID to which each device belongs. It removes the DEVICE ID from the data packet and updates it with CHANNEL ID and the WRITE API to establish an MQTT connection and publish data to the channel.

Typical spectral response variation of a single wavelength (410 nm) can be seen in Fig. 2.11. MATLAB Visualization was used to find the spectral signature of the end node target shown in the interface in Fig. 2.11 (5). Refer to Appendix section 1 for the MATLAB NDVI calculation program.

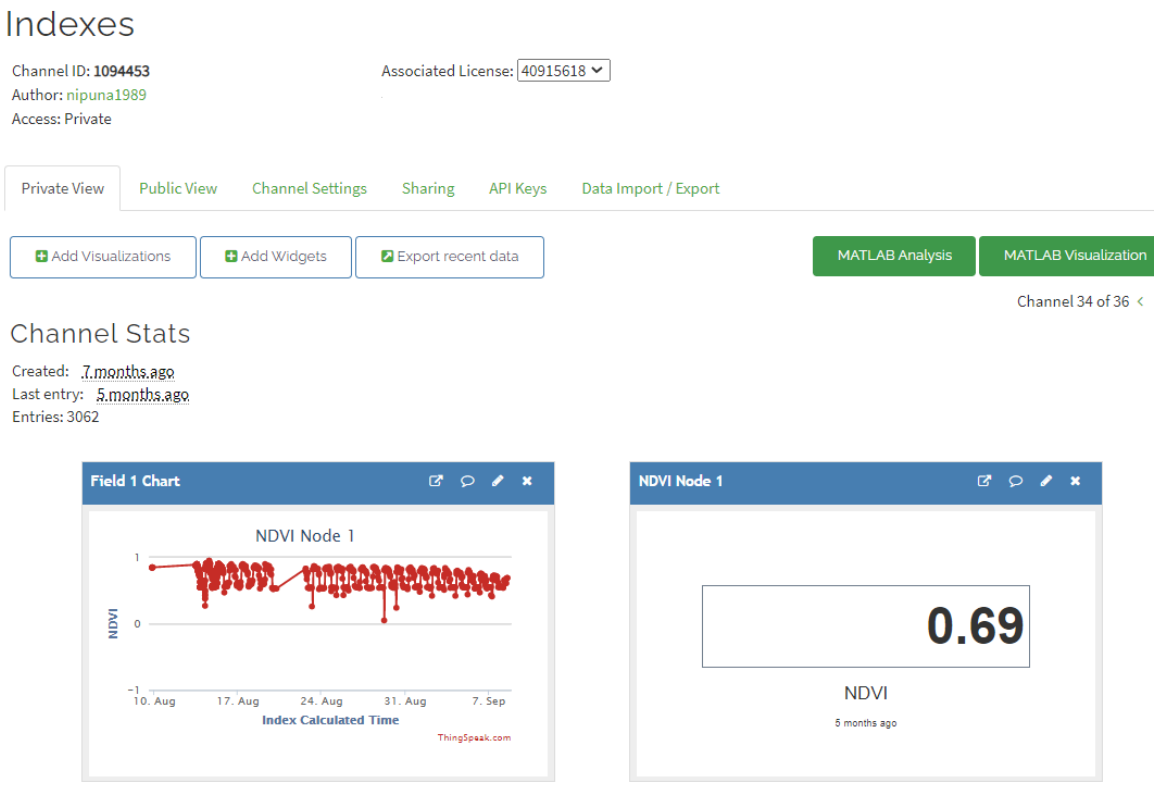

Fig. 2.12 Node 1 vegetation indices visualization
from 10th August to 10th September in 2020

Real-time vegetation indices were calculated by the MATLAB analysis tool and the indices values were updated to channel field (Fig. 2.12) Vegetation indices calculation.

The tested method to calculate the vegetation indices used the one-hour average spectral responses of relevant wavelengths for the index. We like to suggest embedding two spectral sensors in a single node as one sensor to record the incoming and the other to record the reflectance will be the best to calculate real-time vegetation indices in a similar system.

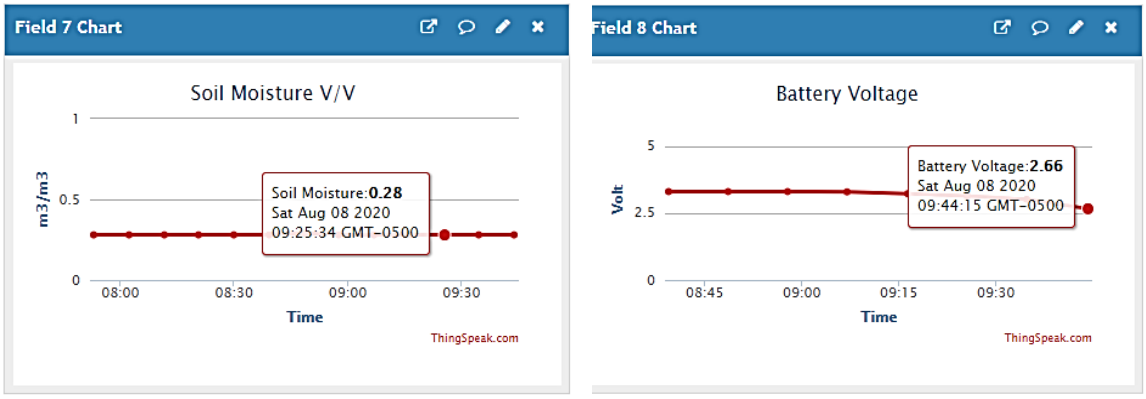

Fig. 2.13 A) Soil moisture telemetry data B) Battery voltage telemetry data

In each channel, there were separate fields for soil and battery voltage telemetry data visualization as shown in Fig. 2.13. Battery voltage telemetry (Fig. 2.12 B) was used to understand the energy usage of the system. This was a help to optimize the node energy consumption during this development process. Battery capacity determination, duty cycle determination was done based on the data collected through the battery voltage telemetry.

2.5 IoT Enabled Sensor Network Field Testing

2.5.1 Location

The experimental setup was established in the field phenotyping facility mentioned in (Bai, et al., 2019), located in the ENREC (Eastern Nebraska Research and Extension Center) of UNL near MEAD, Nebraska, USA. The site GPS Coordinates are latitude 41.15 longitude -96.44 with an area around 6 acres.

The sensor network consisted of ten IOT-enabled sensor nodes (location, number 1 to 10, Fig. 2.14) with down-looking spectroscopy sensors, soil water content sensor and RGB imaging sensor, one IOT-enabled up-looking spectroscopy sensor (location number 11 in Figure 2), and a LoRa gateway (location number 12 in Figure 2).

The initial feasibility study conducted in the same research field, in 2020 early May, to measure the LoRa data transmission with an indoor LoRa gateway and a node at 2 m high proved to be effective for 2 km.

2.5.2 Treatments and Measurements

The T1 zone was irrigated with subsurface drip irrigation and the T2 zone was not irrigated (Fig. 2.14). Irrigation treatments were applied to the soybean crop. Telemetry data of volumetric soil water content under the nodes with soil moisture sensor in nodes. Spectral reflectance of vegetation under each node with the spectroscopy sensor in the visible and near-infrared region was collected. RGB images were captured and stored in SD cards were retrieved after the experiment for analysis.

It is important to mention that data collection was interrupted several times due to damages caused to senor node structures with the wind and logistic issues.

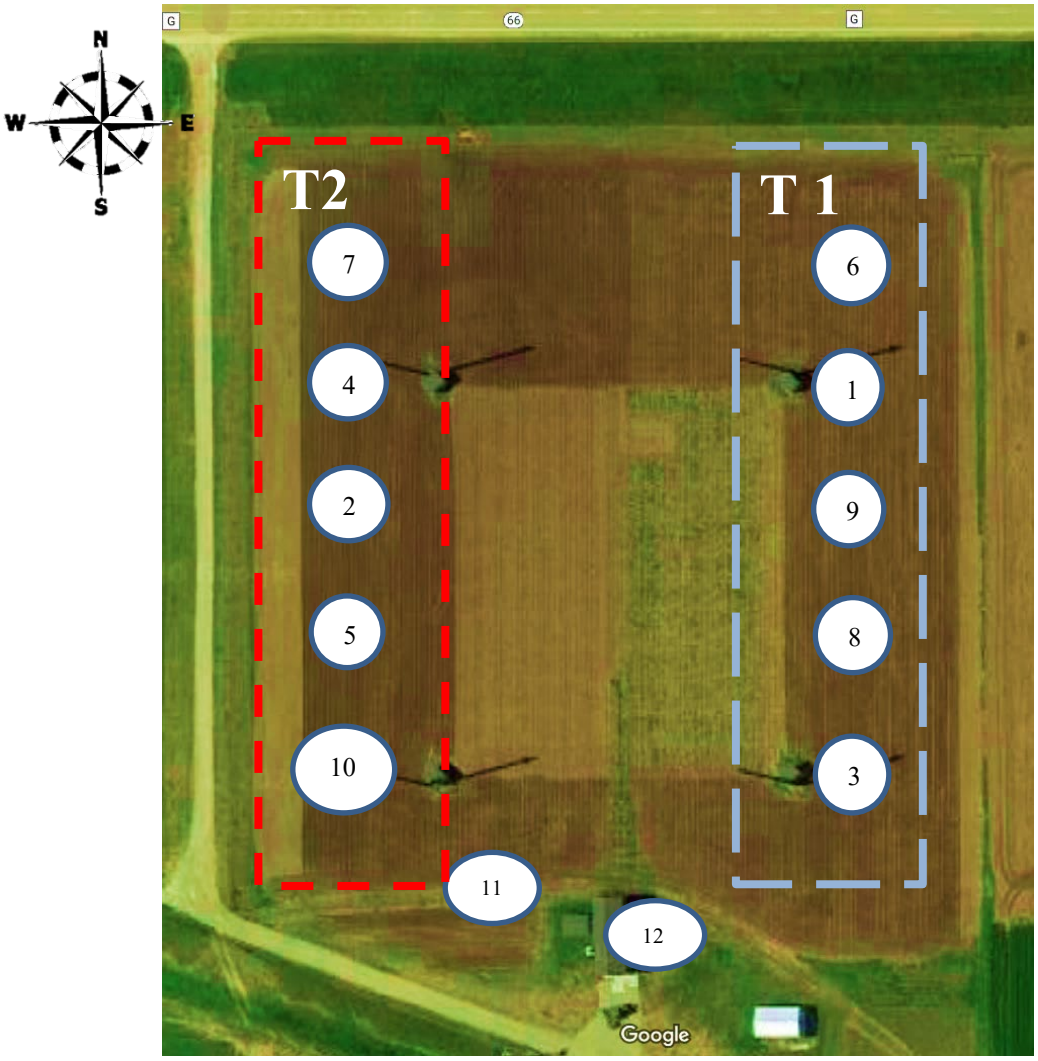

Fig. 2.14 IoT sensor node map and treatments including sensor node locations with node numbers and treatments with treatment numbers
(1 - 10: down looking sensors, 11: up looking sensor, 12: LoRa gateway,
T1: Treatment 1, T2: Treatment 2)

2.5.3 Ground Truth Data Collection

A handheld GreenSeeker sensor (GreenSeeker Handheld Crop Sensor, Trimble Inc) was used to collect NDVI values of the soybean crop under each node. It emits light and measures the reflectance at 660 nm (Red) and 770 nm (NIR) to calculate NDVI (Tremblay, Wang, Ma, Belec, & Vigneault, 2008). The Apogee chlorophyll concentration

meter (Apogee Instruments, Inc.) was used to measure the chlorophyll concentration in absolute units of μmol of chlorophyll per square meter of plant leaf area. Both measurements were collected once every week from 7th August to 14th September.

2.6 Technical Challenges Faced

There were several technical challenges during the research project. Although there were several LoRa development boards on the market, most of them did not support sensor interfacings such as analog sensor readings and I2C communication. It was a major technical difficulty faced during the development stage. Ultimately the Arduino MKR1300 development board was found as the solution with all the expected capabilities.

Operating voltage differences among sensors was also a critical technical issue faced during the development stage. Going through a series of initial sensor testing ultimately directed to select the best combination of sensors.

Although SDI-12 was a standard sensor data communication protocol for multiple sensors, the Arduino MKR1300 did not support this protocol. This issue blocked us by integrating the soil sensor with soil temperature, soil water content, and soil conductivity sensor to the sensor node.

Keep the system in stable condition was the main issue we faced at the implementation and testing stages. The limited ground anchoring depth due to subsurface irrigation was the reason for this issue. Sensor nodes fell due to high wind conditions and it causes severe damages to soil moisture sensor cables and continued data logging.

Rodents attacked some soil moisture sensor cables and solar power cables exposed to the outside. These errors initially misguided the fault identification process of IoT nodes but with time it became easy to identify the errors caused by this issue.

2.7 Conclusion

Sensor selection, data processing unit selection, communication system selection, node energy management cloud data analytics platform selection, and integration of selected modules to a single system was successful. Each part was integrated to make a successful sensor node. There was an initial goal to transmit images through the sensor network. Past studies revealed it is possible even through the LoRa protocol. But that could not be accomplished without violating LoRaWAN fair use policies (Industries, 2021). Therefore, the introduction of an edge image processing capability is the best solution to evaluate RGB image-based data. Except for this goal the entire sensor network setup could be considered as successful.

CHAPTER 3 RESULTS AND DISCUSSION

3.1 Introduction

In this chapter research results were analyzed and discussed in three areas. First, the LoRaWAN network performance was discussed by comparing the number of messages generated by the network nodes with the successfully transmitted messages to the cloud. Second, spectroscopic sensor accuracy was compared with a PAR sensor by comparing the incoming solar radiation measured by the two sensors. Third, manually collected vegetation indices were compared with the data collected by the sensor network from July to August 2020.

3.2 LoRaWAN Network Performance Evaluation

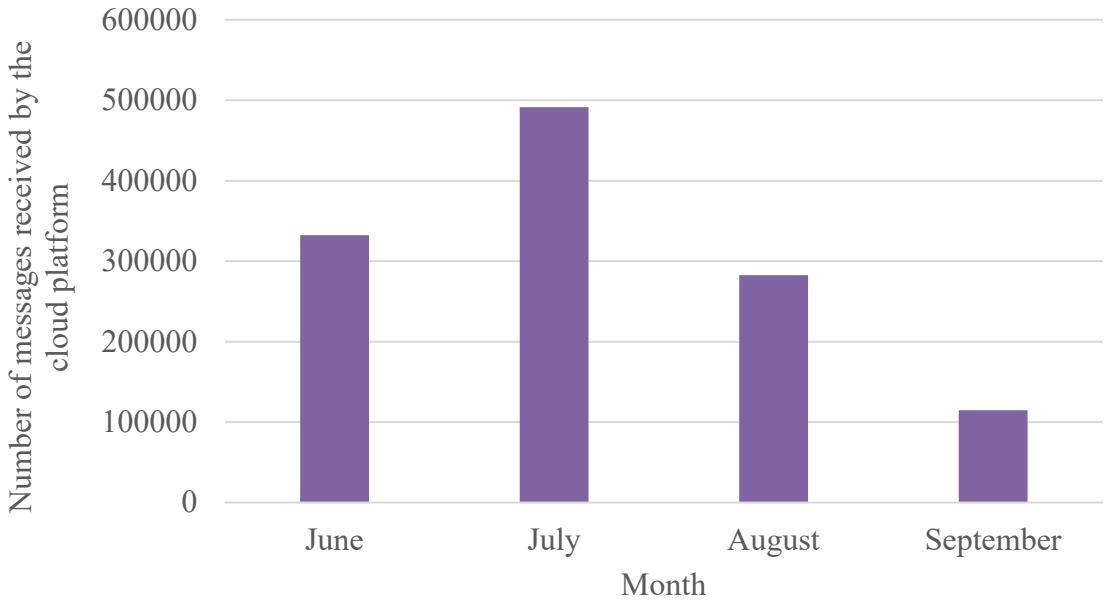

Fig. 3.1 Total number of messages received by the ThingSpeack IoT platform via the LoRa gateway during June 2020 and September 2020

Fig. 3.1 presents the total number of messages received by the ThingSpeak cloud platform. During the experiment period, the maximum number of messages was transmitted in July and the message rate was around 672 messages per hour. In 2020 June, not all 10 sensor nodes were set up yet and in 2020 August and September, some sensor nodes were damaged due to high wind and rain. Therefore, the total number of messages received by the cloud was less than 350 000. The data collection was stopped from 14th September, and it resulted in a low total message count in September.

When the data packets do not reach the planned destination, it is considered as the data packet loss. Radiofrequency interference (RFI), radio signals that are too weak due to distance or multi-path fading, faulty networking hardware, or faulty network drivers, network congestion are some common reasons for wireless data packet loss. The nodes used in this research followed QoS 0 protocol to send data from the nodes to the gateway. It is simply a fire-and-forget method. It is important to find the percentage of data packets transmitted successfully to the cloud under the proposed method to rate the success and find methods to reduce data packet loss.

The distance between the sensor nodes in the field and the gateway varies between 50 m to 150 m (Fig. 2.14). Also, there were no barriers present in the open field. Therefore, there could not be an issue with signal attenuation and very little multi-path fading that may cause data packet loss.

Fig. 3.2 was generated by plotting the total number of messages aired by the active nodes in a selected hour with a total number of messages received by the ThingSpeak cloud. The twelve instances were randomly selected. The lowest number of

messages aired when a single node was activated, and the total message count was 60 messages per hour while the maximum was reported as 794 messages per hour when all the nodes were active including 10 down-looking sensor nodes and 1 up-looking sensor node. We assumed that there was no data loss between the gateway and the cloud platform because gateway used Wi-Fi standard 802.11 b/g/n which is a very efficient data transmission protocols due to carrier-sense multiple access with collision avoidance.

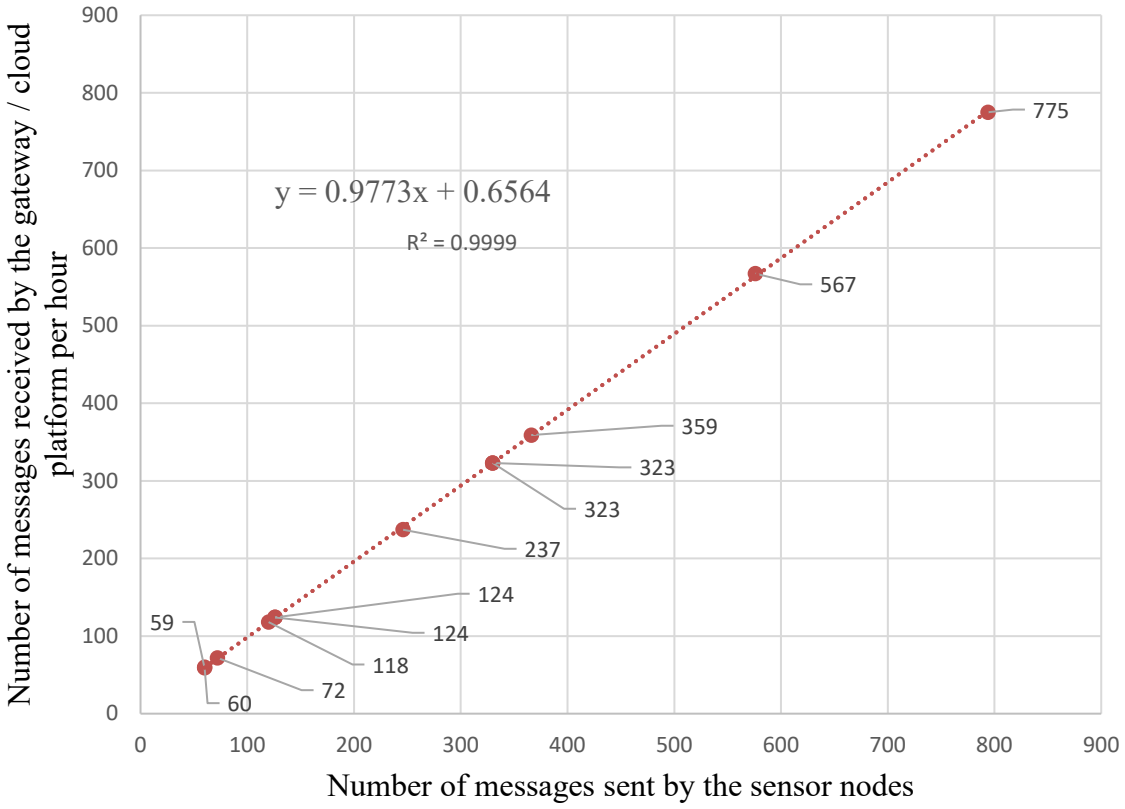

Fig. 3.2 Number of messages received vs number of messages transmitted by the senor nodes in a single hour

Figure 3.2 revealed that the data loss is less than 2.5% up to 800 messages per hour. R squared value is close to 1. The transmission efficiency in Fig. 3.3 was the percentage of successful messages transmitted.

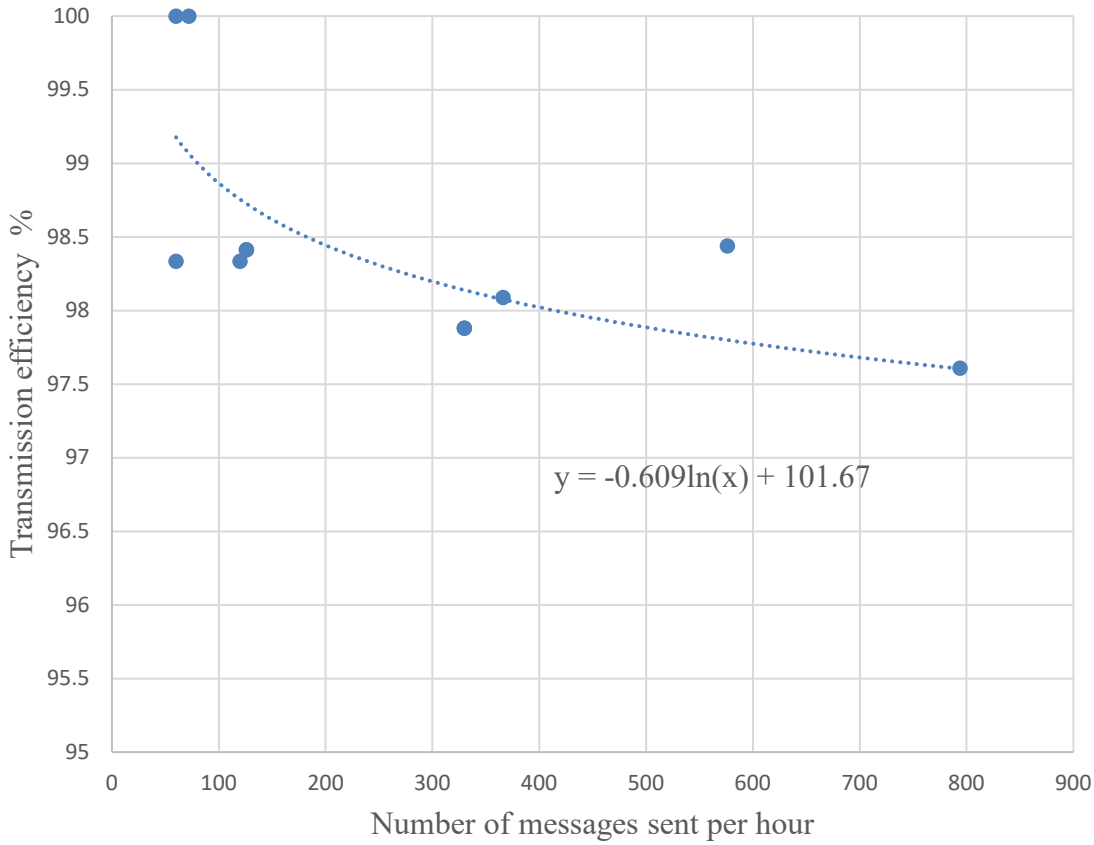

Fig. 3.3 Transmission efficiency vs number of messages sent per hour in the IoT network

For low message rates such as below 100 messages per hour, the data transmission efficiency was almost 100% but efficiency gradually reduced when the message rate increased. In the current experiment, the message transmission rate did not go below 97.5%. Since this experiment was held in a rural farming area and low RFI may be caused by high message efficiency.

3.3 Physical Sensor Node Design Evaluation and Novel Node Design Proposal

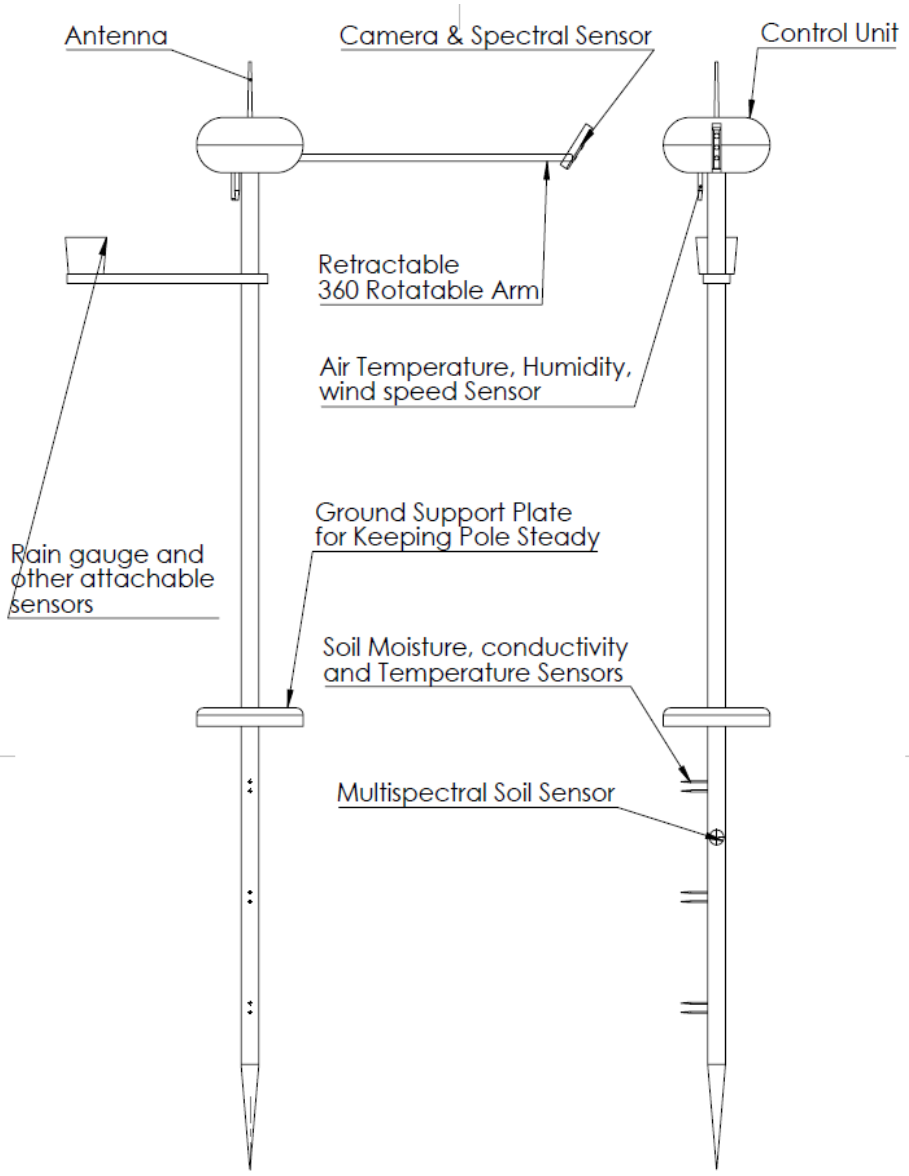

Fig. 3.4 Proposed structural design for the sensor node to improve the current design (Design front view (left) and side view (right))

Sensor node structural design needs to be improved to survive in a long term under field conditions. Extreme wind and rain cause failures in the system. Sensor nodes

fell often due to wind and loosen soil due to rain several times. Images collected during windy situations were blurred. Sometimes insects covered the spectroscopic sensor and imaging sensor. Rodents caused damages to soil moisture sensor wires and solar power cables.

Fig. 3.4 contains the improved node design. In this design to reduce the wind force on the node effective force area was reduced and included a retractable arm for cameras and spectroscopic sensor. The proposed design has no open cables that attract rodents. Further, it has 2 degrees of freedom for the camera and spectroscopic sensor to increase the target area. The author proposed to have above-ground and below-ground parts that can be detachable to allow farmers to maneuver the machinery such as tractors over the crops easily.

3.4 Spectroscopic Sensor Validation

Spectroscopic sensor evaluation was the second objective in this research to validate the data collection process.

3.4.1 In-house Testing

Before the in-field sensor testing, in-house sensor testing was conducted. The inbuilt light sources in the Spark fun triad spectroscopy-sensor-board were used as the light source. A white reference standard was used for calibration and calculated the CF_X values (wavelength specific correction factor). The sensor was placed on an 8 mm standoffs and it helped to keep the distance between the sensor and the target at 8 mm constant height.

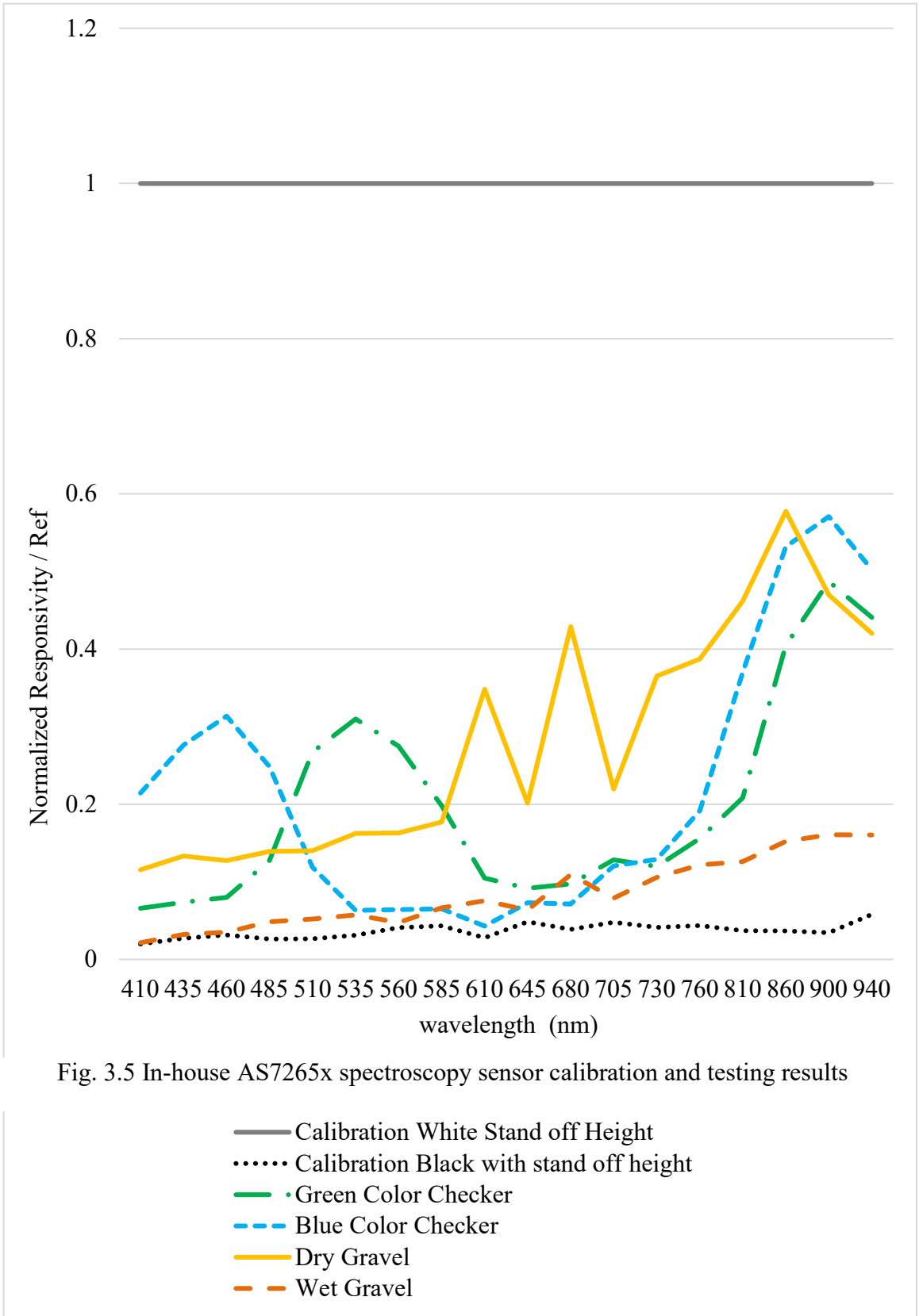

Fig. 3.5 In-house AS7265x spectroscopy sensor calibration and testing results

averaged. Parallax Data Acquisition tool (PLX-DAQ) (Parallax Inc.) was used to transfer data from Arduino development board to excel. Fig. 3.5 represents the results of the standard calibration panel, standard black calibration panel, standard ColorChecker blue (X-Rite, Inc.), standard ColorChecker green, wet gravel, and dry gravel. As expected, blue reflectance was at a maximum at the wavelength 460 nm, green gave a peak reflectance was at a maximum at the wavelength 530 nm, dry gravel (soil) had the maximum at 860 nm, but the reflectance was much higher in the NIR range than the visible spectrum. Since wet gravel was a dark color, wet gravel had less reflectance in the visible spectrum. Furthermore, water in wet gravel absorbed more NIR radiation and it was the reason for the lower reflectance in the NIR region of the wet gravel sample as shown in Fig. 3.5.

3.4.2 Spectroscopic Sensor Accuracy Validation with LI-190R Quantum Sensor

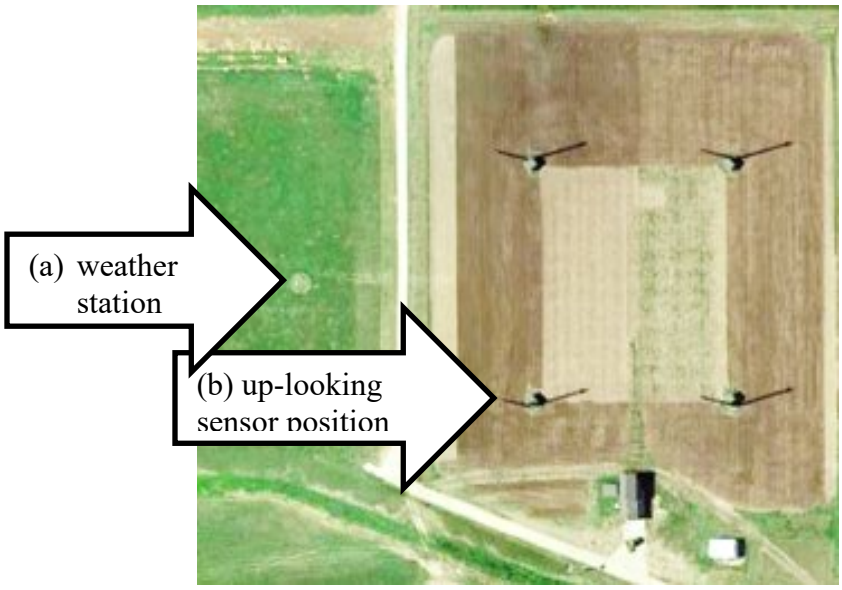

Fig. 3.6 Location map of the PAR sensor and the up-looking sensor in the experimental site (a) weather station (b) up-looking sensor position.

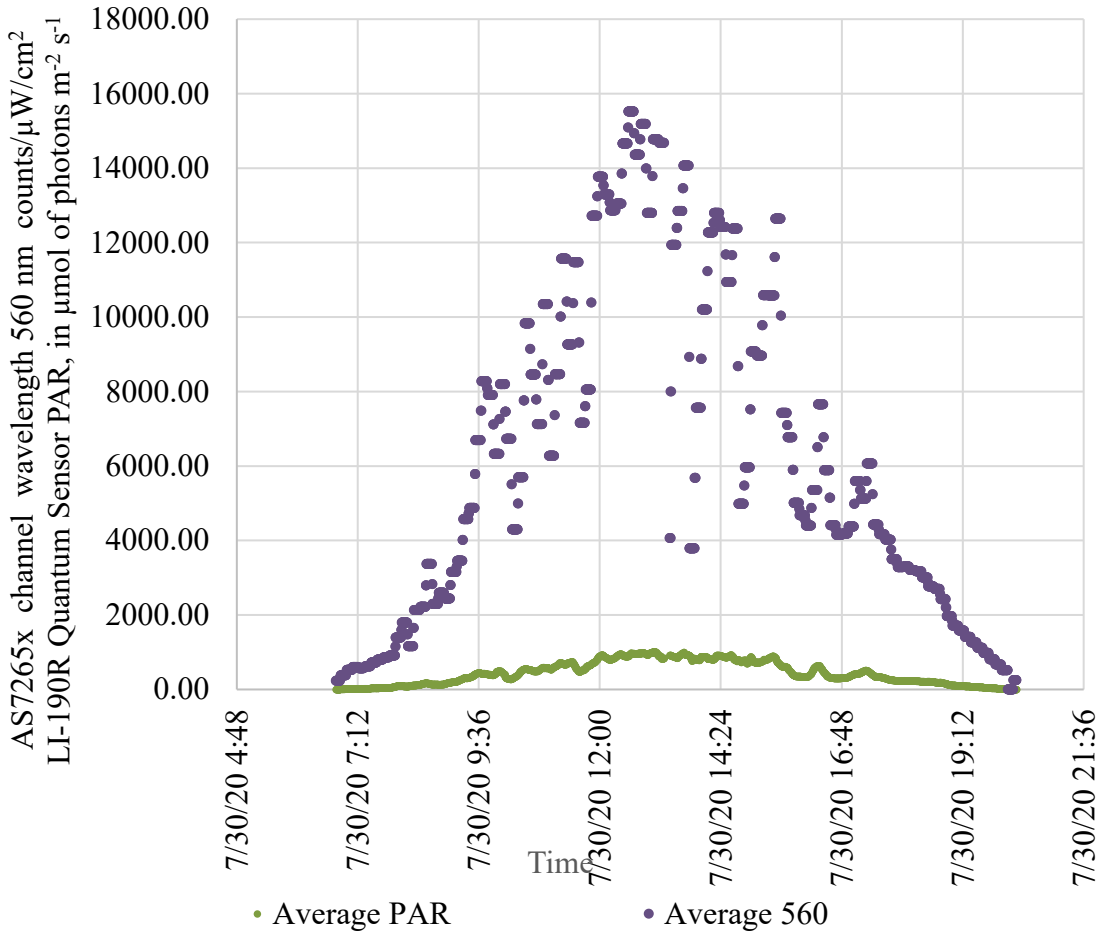

Fig. 3.7 PAR Sensor output vs up-looking spectral sensor wavelength 560 nm output.

The up-looking sensor calibration was important to verify the accuracy of the AS72s65x spectral sensor. A Photosynthetically Active Radiation (PAR) sensor (LI-190R Quantum Sensor, LI-COR, Inc.) in a nearby weather station (Fig. 3.6) was used for this purpose. The PAR sensor logged incoming solar radiation every 5 minutes. Since the AS7265x up-looking sensor took 10 minutes for a single data logging cycle, the moving average of two sensors for a 10-minute interval was calculated and plotted in Fig. 3.7.

The Fig. 3.7 graph compared wavelength was 560 nm in the spectral sensor with PAR sensor output. The reason to select 560 nm for comparison was that it is the peak of the incoming solar radiation wavelength to the earth's surface.

Fig. 3.7 clearly shows that the AS7265x spectral sensor at wavelength 560 nm, followed the diurnal pattern of the incoming radiation variation similar to the PAR sensor. It is important to note that the day 7/30/2020 was cloudy. Sunlight was fluctuated quite a lot caused by frequent cloud movements and it ccould be noticed in Fig. 3.7.

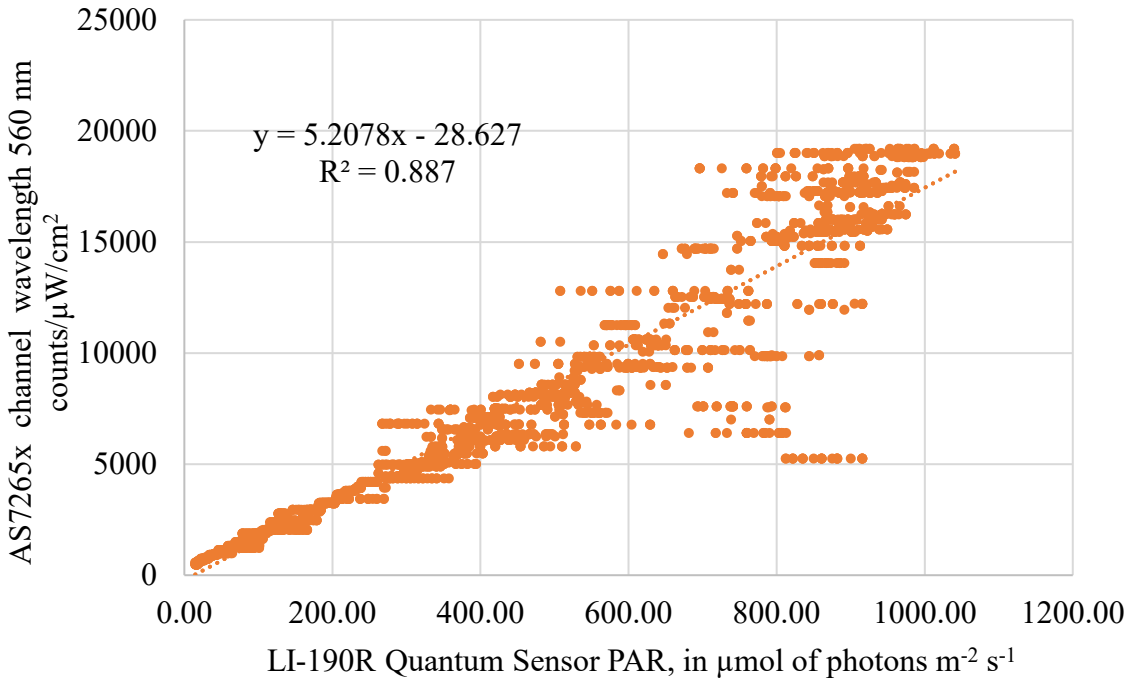

Fig. 3.8 AS7265x wavelength 560 nm output vs LI-190R quantum sensor PAR in 1- to -1 graph

The relationship between the two sensors was linear with $R^2 = 0.887$ (Fig. 3.8). According to the AS7265x datasheet (AG, 2018), the accuracy of the channel in counts/µW/cm^2 is ±12%. Therefore the developed up-looking sensor performed reasonably in measuring the incoming solar radiation in the PAR region.

Fig. 3.8 revealed that sensor accuracy becomes low after PAR sensor reading exceeded 400 µmol of photons m^{-2} s^{-1}.

3.5 Diurnal and Seasonal NDVI Variation

The NDVI value was calculated by the MATLAB simulation and visualization option on the ThingSpeak. We first averaged two reflectance values received per hour and then displayed them on the ThingSpeak dashboard. Since the spectral sensor single reading taking 1 hour to upload under the method, the vegetation index values shown in the application layer have around 1-hour lag. This issue can be improved by simply encapsulating all the sensor readings into a single message. This feature was not activated in this research because we initiated the research with the trial version of the MATLAB ThingSpeak which only allowed 1 channel to be updated by one message at a time.

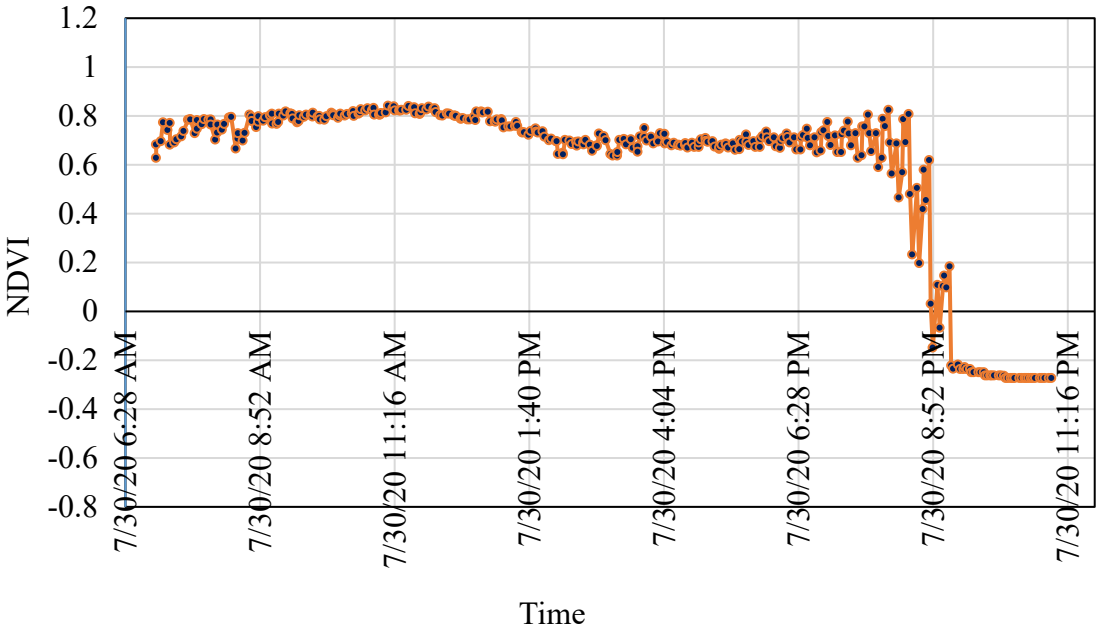

Fig. 3.9 Diurnal NDVI moving average value in the node

Fig. 3.9 shows the diurnal NDVI variation of the vegetation under Node 1. NDVI value reached a peak in the morning between 9-10 am and reduced at the noon and again

reach a peak in the evening between 4-6 pm. This phenomenon was observed by (Beneduzzi, Souza, Bazzi, & Schenatto, 2017) and (Kim, Glenn, Park, Ngugi, & Lehman, 2012). It is important to note that the soybean leaves wilt under high sunlight conditions and high-temperature conditions. This phenomenon was explained by (Setiyono, Weiss, Specht, Cassman, & Dobermann, 2008) as a biological process by soybean to save plant water under stressed climatic conditions. Since the NDVI is related to crop canopy area as well as leaf water content and leaf structure the average diurnal NDVI pattern that appeared in our research seemed reasonable. However, to improve the efficiency of the message transmission efficiency the node design could be improved by including both up-looking and down-looking sensor in the same sensor node. This feature was embedded in the SRS Spectral Reflectance Sensor manufactured by (METER Group, SRS Spectral Reflectance Sensor Operator's Manual, 2020). This product has two sensors named the hemispherical version of the sensor and the field Stop Version of the sensor, where the hemispherical version is the up-looking sensor while the field Stop Version is the target locked sensor. Since the focus of this research was to develop and

evaluate the IoT sensor system with IoT data analysis capability in-depth, NDVI value optimization was not studied.

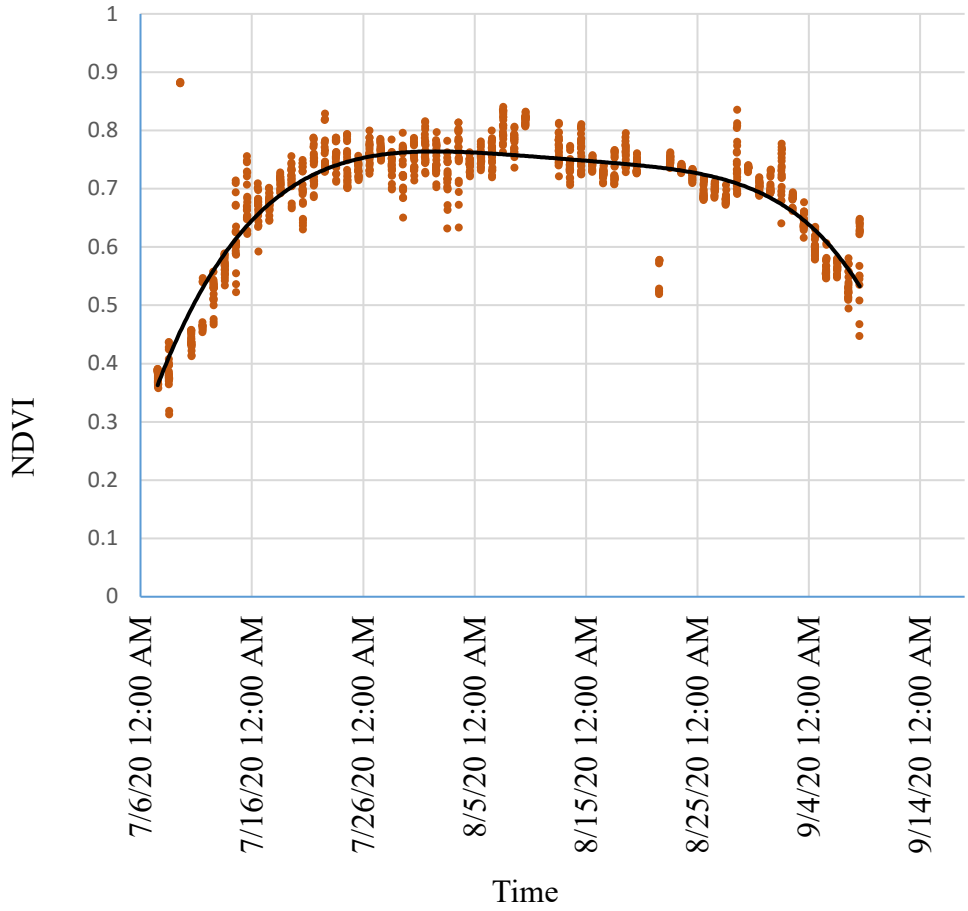

Fig. 3.10 Seasonal peak NDVI with polynomial trendline (order 6) for Node 1

Fig. 3.10 shows the seasonal NDVI variation and the daily peak NDVI values were used to generate this graph after removing certain outliers. The decline of the NDVI value after the soybean reached the R7 stage could be seen in Fig. 3.10. At the early season such as, in early June the NDVI was low because the crop canopy coverage was low. The images collected by the sensor node verified that conclusion. Only one sensor node was able to capture the entire growing season.

3.6 The capability of IoT Sensor Network to Distinguish the Two Treatments.

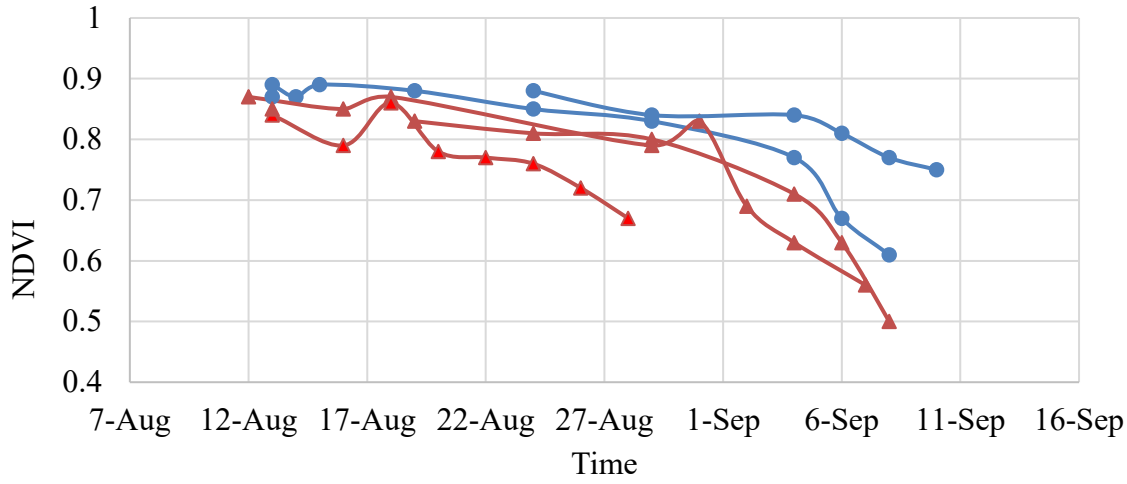

Fig. 3.11 NDVI variation comparison between treatments

Fig. 3.11 shows the NDVI values of node 1 and node 8 which were irrigated and node 4, node 7, and node 10 which were from non-irrigated treatment.

Even without statistical analysis, NDVI differences in the two treatments could be identified. But further studies are required to validate the results. The main reason for avoiding the July 2020 data was that the MATLAB NDVI calculation algorithm (Appendix) was not developed until early August. However, due to frequent rain in early August, the effect of irrigation could not be identified from the plants. The rain was enough to meet the requirements of water use for soybean. Only in late August and early September, the effect could be identified.

3.7 Manual NDVI Value vs Sensor Node NDVI Comparison

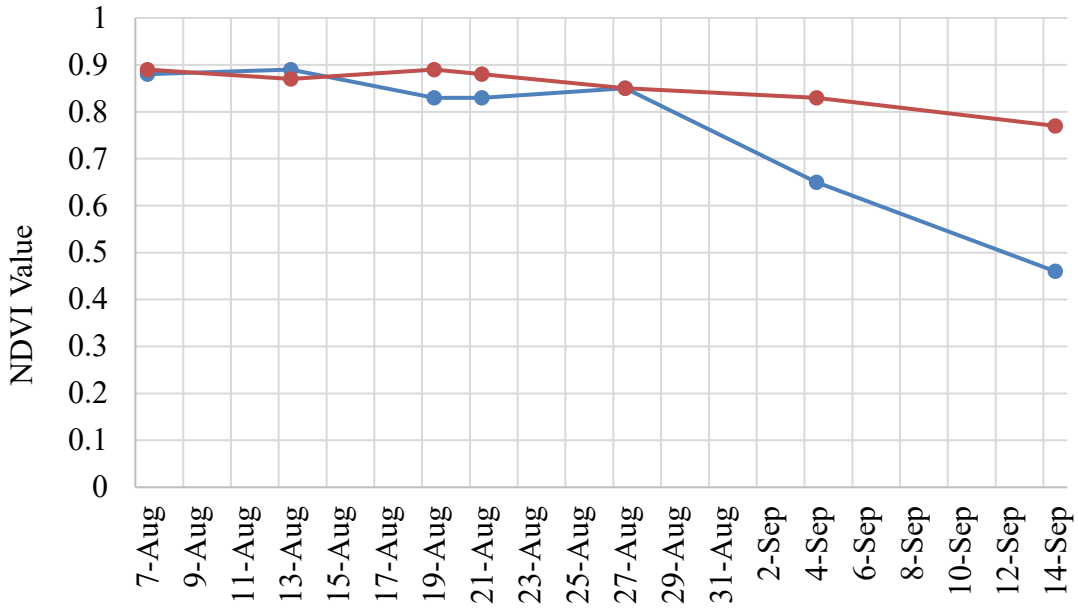

Fig. 3.13 Manual NDVI Data and IoT Node NDVI Data Comparison Node 1

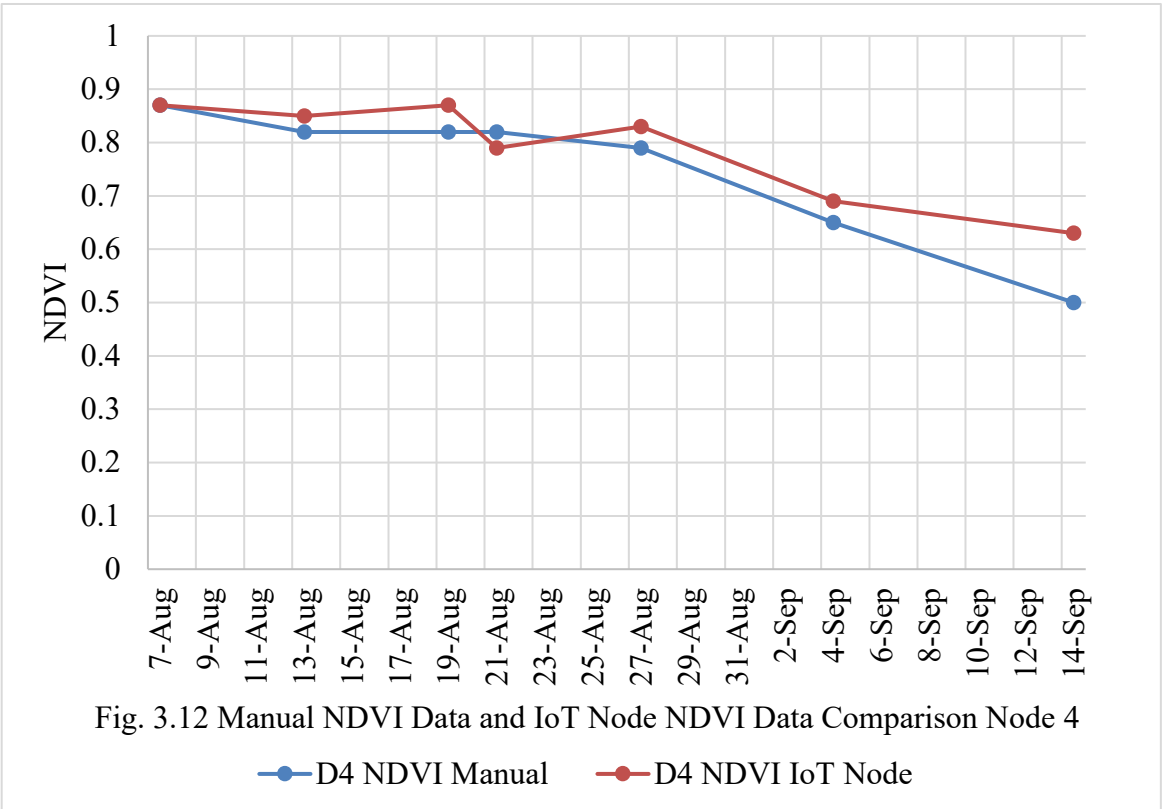

Fig. 3.12 Manual NDVI Data and IoT Node NDVI Data Comparison Node 4

Fig. 3.12 and Fig. 3.13 are a comparison between manually collected NDVI by the handheld green seeker with IoT sensor NDVI data.

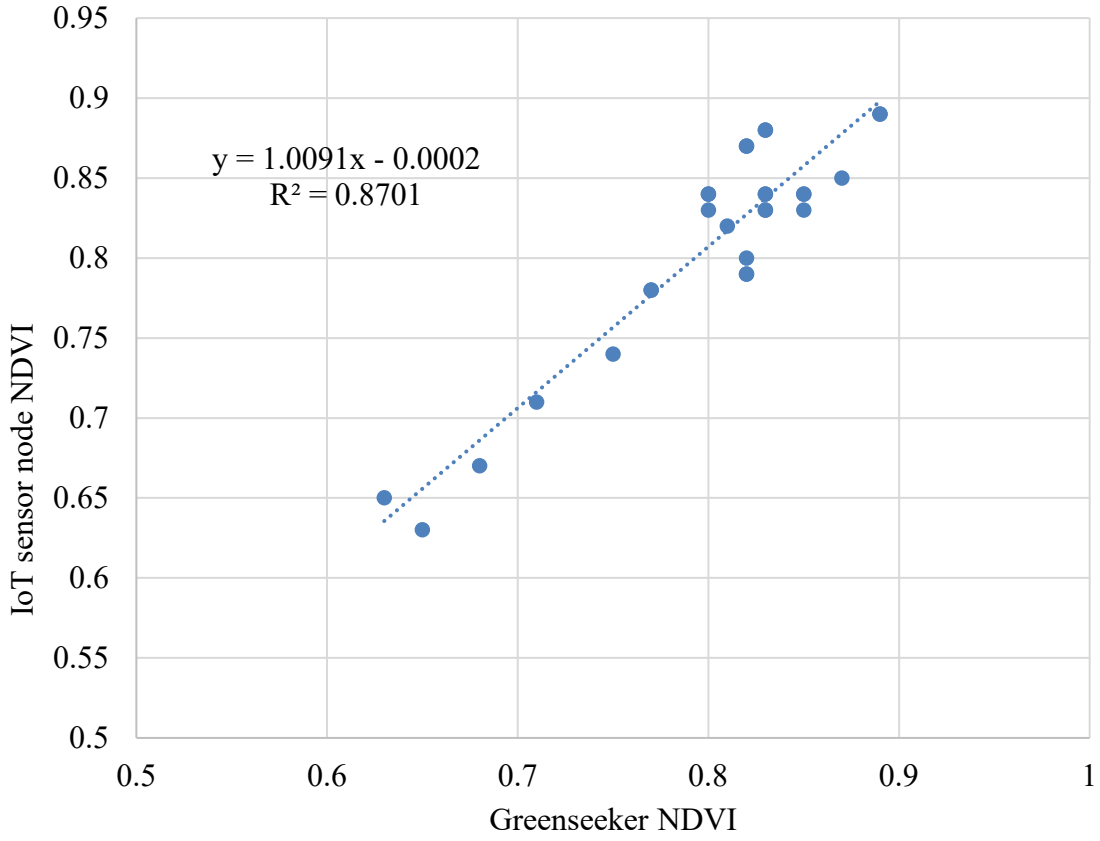

Fig. 3.14 Manually collected NDVI vs IoT sensor node collected NDVI between August 7th and September 9

Twenty-eight data points were selected from both manually collected NDVI data and IoT sensor node collected NDVI values to plot the 1:1 graph in Fig. 3.14 with the same timestamp. The two sensors have a coefficient of determination value of around 0.87. This correlation revealed that the data collected by the nodes had a reasonable accuracy with the Green seeker handheld sensor. However, the Greenseeker is an active NDVI measuring instrument that does not depend on the sunlight.

These results verify the capability of the spectral reflectance sensors on the IoT sensor network could monitor and record the daily and seasonal variations of NDVI, which is an important indicator of plant growth status and plant stress conditions.

3.8 Competitive Advantages of the Novel Sensor Network Over Commercial Solutions

Fig. 3.15 Commercial meters for crop parameter measurement based on vegetation indices (a) Apogee chlorophyll concentration meter (b) SPAD 502 plus chlorophyll meter (c) atLEAF Chlorophyll meter (d) SRS - Spectral Reflectance Sensor

To the best of the authors' knowledge there are commonly used three manual handheld instruments available to study plant chlorophyll concentration (Fig. 3.15 (a), (b) and (c)) and one in field NDVI and PRI reflectance sensor (Fig. 3.15 (d)) which was used by farmers and researchers for assessing canopy variables such as light use

efficiency (LUE), biomass and crop yield, crop and forest phenology, canopy growth, and photosynthetic performance/CO_2 uptake.

The handheld Green seeker sensor, which measures NDVI value with two wavelengths: 656 nm and 774 nm, are used as a nitrogen fertilizer requirement analyzing tool. SPAD 502 Plus Chlorophyll Meter measures the plant leaf chlorophyll content based on wavelengths 650 nm and 940 nm and this is an active sensor and has two LEDs that emit light with high intensity in the 650 nm and 940 nm wavelengths. The output of the meter is in terms of SPAD unit, ranging from 0 to 50. The advantages of this device were that scientists had used it for decades and had experimental result-based forecasting equations to predict leaf nitrogen status and chlorophyll status for different plant species. The disadvantages were that the measurements need to be taken on individual plant leaves and need several samples from a single plant to estimate the overall plant nitrogen requirement and estimate chlorophyll concentration.

Chlorophyll Concentration Meter, Model MC-100 by Apogee Instruments, INC used 653 nm and 931 nm as the operation wavelengths. The Apogee chlorophyll concentration meter is calibrated to measure chlorophyll concentration in leaves with units of μmol of chlorophyll per m^2. It indicated the leave chlorophyll concentration in Chlorophyll Concentration Index (CCI). Further, the device can output the chlorophyll concentration with a SPAD unit. Measurements of the device can be converted between SPAD unit and CCI value and it is one of the main advantages of this device. The disadvantages of this device are the same as the SPAD 502 meter.

atLEAF Chlorophyll meter estimates the chlorophyll concentration based on wavelengths at 640 nm and 940 nm. The output was defined using atLEAF units which has a span between 0 and 99.9. This was a similar product like SPAD 502 and Chlorophyll Concentration Meter, Model MC-100.

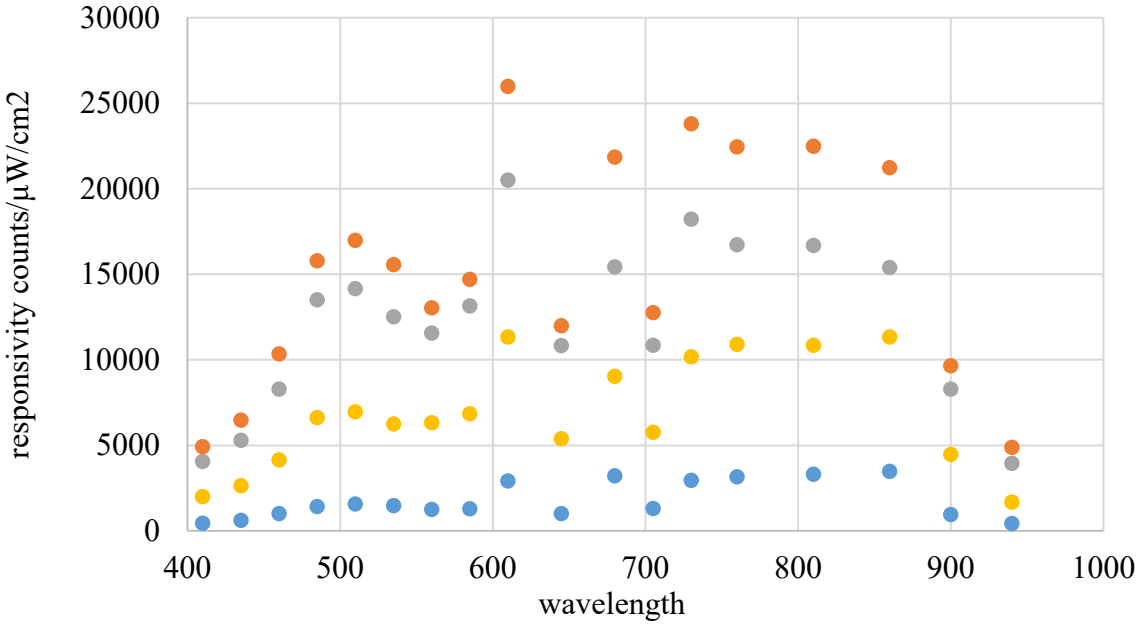

Fig. 3.16 AS72s65x response to incoming solar radiation in four differentnt timestamps within a day

All the handheld chlorophyll concentration meters have the same wavelength range for measurements, and they are active sensors. They were 640 - 655 nm and 930 – 940 nm range. The AS72s65x sensor board has 18 channels with 18 different wavelengths (Fig. 3.16). Because of that, it is possible to develop a novel handheld vegetation index sensor with multiple capabilities compared to the existing handheld devices.

As an infield sensor, SRS sensor (Fig. 3.15) with NDVI wavebands: 650 and 810 nm central wavelengths, with 10 nm full width half maximum bandwidths and PRI

wavebands: 532 and 570 nm central wavelengths, with 10 nm full width half maximum bandwidths capable of collect canopy level VIs. Since the sensor has a large field of view, the output can be related to canopy level parameters.

Compared to SRS sensor the developed solution has the advantage of collecting entire canopy spectral signature in 18 wavelengths.

For canopy level parameter studies and precision agricultural practices, in-field sensors are advantageous compared to handheld devices. The advantages are continuous target-oriented data collection, high accuracy, low recurrent cost for labor, less data preparation time, and less data analytics and presentation time.

CHAPTER 4 GENERAL CONCLUSIONS AND WAY FORWARD

In agriculture, real-time sensing is crucial for decision-making. It is also an essential part of precision agricultural management. Soil, crop, and climatic parameters are the three main categories required for farmers to make decisions. Scientists who study natural resources and ecology are also interested in real-time data collection for soil, water, and vegetation. The spectral signature of an object or a material is a unique feature. The spectral signature can infer the status of soil nutrient, soil water content, crop health status, crop growth stage, and crop nutrient deficiency. This technology has demonstrated its advantages for decades. But the cost is an issue that limits the use of this technology for both scientific and agricultural purposes. However, with the introduction of low-cost multispectral sensors, the possibility of developing new solutions to use in agricultural and natural resource monitoring becomes feasible. It will open a new frontier in precision crop management. Sensor-based data collection has advantages such as fast data collection, easy data analytics, low labor involvement, reliability, and high accuracy. Parameter sensing and conversion to data, data transmission, data storing, data analyzing, and data visualization are functionalities for a decision-making system.

Internet of things has the potential to integrate all the functionalities stated above. State-of-the-art technologies such as infrastructure (ex: - cloud data storage), communication technologies (ex: - wired and wireless data transmission), data visualization techniques (ex:- augmentation reality glasses), and data analyzing techniques (ex:- machine learning) are compatible with IoT.

This research successfully integrated both low-cost multispectral sensors with IoT technology to create a novel sensor network. The image transmission via LoRa was not successful due to the fair use policy implementation. However, the imaging sensor successfully captured and saved images. The soil moisture data collected was not able to demonstrate the complete season soil moisture variation due to frequent changes of soil moisture sensor location change. However, the soil moisture readings were successfully transmitted during the whole growing season.

This sensor network can capture spectral signatures and generate real-time vegetation indices that are important for crop management. The results concluded that the LoRa communication protocol and LoRaWAN technology have the feasibility to set up wide-range low throughput wireless sensor networks. Furthermore, AS7265x is a low-cost sensor module with reasonable accuracy while it was able to continuously capture the spectral signature of soybean within a growing season. Also, this research demonstrated the feasibility of using the spectroscopic sensor for soybean irrigation requirement identification and growth stage identification.

Structural, ergonomic, and functional improvements need to fulfill in the future. The wind-resistant structure is an important design consideration. Two-way communication is required to make changes in internal parameters easily. The introduction of other established vegetation indices to the system is a required functionality.

LoRaWAN does not support transmitting high-quality image data but literature shows the possibility of transmitting low quality images. However, the LoRaWAN is not

recommended to transmit image data. The reason is it is a license free bandwidth and according to fair use policy the uplink airtime is limited to 30 seconds per day (24 hours) per node and the downlink messages are limited to 10 messages per day (24 hours) per node according to the Things Network (Industries, Duty Cycle, 2020). Therefore, the sensor node must have the capability in implementing image processing to derive information from the images. This information can be insect density in the crop canopy, disease infection rate, flowering stage recognition, and physical damages to plants. This data can be forwarded to the cloud as telemetry data to the cloud platform for store and visualize. Then farmers may be able to use these data for yield improvements on their farms in the future.

REFERENCES

A. Gitelson, A., Gritz, Y., & N. Merzlyak, M. (2002). Relationships between leaf chlorophyll content and spectral reflectance and algorithms for non-destructive chlorophyll assessment in higher plant leaves. *Journal of Plant Physiology*, 271–282. doi:https://doi.org/10.1078/0176-1617-00887

ABBATE, J. (2000). *Inventing the Internet.* Cambridge, Massachusetts: MIT Press.

AG, a. (2018). *AS7265x Smart 18-Channel VIS to NIR Spectral_ID 3-Sensor Chipset with Electronic Shutter.* ams AG. Retrieved from https://ams.com

AliKhattak, H., AliShah, M., Khan, S., Ali, I., & Imran, M. (2019). Perception layer security in Internet of Things. *Future Generation Computer Systems, 100*, 144-164. doi:https://doi.org/10.1016/j.future.2019.04.038

Al-Sarawi, S., Anbar, M., Alieyan, K., & Alzubaidi, M. (2017). Internet of Things (IoT) Communication Protocols: Review. (pp. 685-690). Amman, Jordan: IEEE. doi:10.1109/ICITECH.2017.8079928

Anghelof, M. M., Suciu, G., Craciunescu, R., & Marghescu, C. (2020). Intelligent System for Precision Agriculture. *2020 13th International Conference on Communications (COMM)* (pp. 407-410). IEEE. doi:10.1109/COMM48946.2020.9141981

Ashton, K. (2009). That ‘internet of things’ thing. *RFID Journal*, 97-114.

Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, 2787-2805.

Bai, G., Ge, Y., Scoby, D., Leavitt, B., Stoerger, V., Kirchgessner, N., . . . Awada, T. (2019). NU-Spidercam: A large-scale, cable-driven, integrated sensing and robotic system for advanced phenotyping, remote sensing, and agronomic research. *Computers and Electronics in Agriculture, 160*, 71-81. doi:https://doi.org/10.1016/j.compag.2019.03.009

Beneduzzi, H., Souza, E., Bazzi, C., & Schenatto, K. (2017). Temporal Variability In Active Reflectance Sensor-Measured Ndvi In Soybean And Wheat Crops. *Journal of the Brazilian Association of Agricultural Engineering*, 771-781. doi:http://dx.doi.org/10.1590/1809-4430-Eng.Agric.v37n4p771-781/2017

Browning, D. (2018, JULY 25). *IoT Started with a Vending Machine.* Retrieved from Machine Design: https://www.machinedesign.com/automation-iiot/article/21836968/iot-started-with-a-vending-machine

Castellano, G., Deruyck, M., Martens, L., & Joseph, W. (2020). System Assessment of WUSN Using NB-IoT UAV-Aided Networks in Potato Crops. *IEEE Access, 8*, 56823-56836. doi:10.1109/ACCESS.2020.2982086

Cohen, D. (1979). On Interconnection of Computer Networks. In K. G. Beauchamp (Ed.), *In Interlinking of Computer Networks* (pp. 175-183). Bonas: Springer. doi:https://doi.org/10.1007/978-94-009-9431-7

Dragino Technology Co., L. (2018). *LG01 LoRa Gateway User Manual.* Shenzhen, China: Shenzhen Dragino technology development co.LTD.

Dragino Technology Co., L. (2019). *LG02 LoRa Gateway User Manual.* Dragino Technology Co., LTD.

Ejaz, W., & Anpalagan, A. (2018). Communication Technologies and Protocols for Internet of Things. *Internet of Things for Smart Cities*, 17-30. doi:https://doi.org/10.1007/978-3-319-95037-2_2

Fan, J., Zhang, Y., Wen, W., Gu, S., Lu, X., & Guo, X. (2020). The future of Internet of Things in agriculture: Plant high-throughput phenotypic platform. *Journal of Cleaner Production, 280 Part 1*. doi:https://doi.org/10.1016/j.jclepro.2020.123651

Gamon, J., Kovalchuck, O., Wong, C., Harris, A., & Garrity, S. (2015). Monitoring seasonal and diurnal changes in photosynthetic pigments with automated PRI and NDVI sensors. *Biogeosciences, 12*, 4149–4159. doi:10.5194/bg-12-4149-2015

Gungor, V., Sahin, D., Kocak, T., Ergut, S., Buccella, C., Cecati, C., & Hancke, G. (2011). Smart Grid Technologies: Communication Technologies and Standards. *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, 7*(4), 529-539. doi:10.1109/TII.2011.2166794

Industries, T. T. (2021, 2 28). *https://www.thethingsnetwork.org/docs/lorawan/duty-cycle*. Retrieved from www.thethingsnetwork.org: https://www.thethingsnetwork.org/docs/lorawan/duty-cycle.html

Jiang, J., & Su, K. (2012). Management Platform Architecture of Modern Tobacco Logistics Based on Internet of Things Technologies. *LISS 2012* (pp. 1403-1409). Berlin: Springer. doi:https://doi.org/10.1007/978-3-642-32054-5_199

Jianjun, Y., Deqin, X., & Shunbin, H. (2014). A Remote Monitoring System for the Process of Crop Growth in Precision Agriculture Based on a Wireless Visual

Sensor Network. *Sensor Letters, 12*, 630-637. doi:https://doi.org/10.1166/sl.2014.3141

Karagiannis, V., Chatzimisios, P., Vazquez-Gallego, F., & Alonso-Zarate, J. (2015). A Survey on Application Layer Protocols for the Internet of Things. *Transaction on IoT and Cloud Computing, 3*(1), 11-17.

Katsoulas, N., Bartzanas, T., Kittas, C., & Tzounis, A. (2017, December). Internet of Things in agriculture, recent advances and future challenges. *Biosystems Engineering, 164*, 31-48. doi:https://doi.org/10.1016/j.biosystemseng.2017.09.007

Kim, Y., Glenn, D., Park, J., Ngugi, H., & Lehman, B. (2012). CHARACTERISTICS OF ACTIVE SPECTRAL SENSOR FOR PLANT SENSING. *Transactions of the ASABE, 55*(1), 293-301.

Leo, M., Battisti, F., Carli, M., & Neri, A. (2014). A federated architecture approach for Internet of Things security. *2014 Euro Med Telco Conference (EMTC)* (pp. 1-5). Naples, Italy: IEEE. doi:10.1109/EMTC.2014.6996632

Libelium. (2019). *New vineyard project developed with Libelium IoT platform on Agrotech, the app for crop management, powered by Efor and Ibercaja on Microsoft Azure.* Zaragoza: Libelium Comunicaciones Distribuidas S.L. Retrieved from https://www.libelium.com/libeliumworld/success-stories/new-vineyard-project-developed-with-libelium-iot-platform-on-agrotech-the-app-for-crop-management-powered-by-efor-and-ibercaja-on-microsoft-azure/

Libelium. (2021, 01 16). Retrieved from https://www.libelium.com/

LinkSmart, I. (2021, 1 22). *Linksmart*. Retrieved 01 22, 2021, from https://linksmart.eu/

METER Group, I. U. (2020). *SRS Spectral Reflectance Sensor Operator's Manual.* Pullman WA: METER Group, Inc. USA. Retrieved from http://manuals.decagon.com/Manuals/14597_SRS_Web.pdf

METER Group, I. U. (2020). *SRS Spectral Reflectance Sensor Operator's Manual.* Pullman WA: METER Group, Inc. USA.

Monteith, J. L. (1977). Climate and the efficiency of crop production in Britain. *Philosophical Transactions Royal Society of London B*, 277-294. doi:https://doi.org/10.1098/rstb.1977.0140

Navani, D., Jain, S., & Nehra, M. (2017). The Internet of Things(IoT) : A Study of Architectural Elements. *13th International Conference on Signal-Image Technology and Internet-Based Systems* (pp. 473-478). Jaipur, India: IEEE. doi:10.1109/SITIS.2017.83

Ngu, A., Gutierrez, M., Metsis, V., Nepal, S., & Sheng, Q. (2017). IoT Middleware: A Survey on Issues and Enabling Technologies. *IEEE INTERNET OF THINGS JOURNAL, 4*(1), 1 - 20. doi:10.1109/JIOT.2016.2615180

Nguy-Robertson, A., Gitelson, A., Peng, Y., Viña, A., Arkebauer, T., & Rundquist, D. (2012). Green Leaf Area Index Estimation in Maize and Soybean: Combining Vegetation Indices to Achieve Maximal Sensitivity. *Agronomy Journal, 104*(5), 1336-1347. doi:https://doi.org/10.2134/agronj2012.0065

Patil, K., & Kale, N. (2016). A Model for Smart Agriculture Using IoT. *2016 International Conference on Global Trends in Signal Processing, Information*

Computing and Communication (ICGTSPICC) (pp. 543-545). IEEE. doi:10.1109/ICGTSPICC.2016.7955360

Ploennigs, J., Cohn, J., & Clark, A. (2018, November 29). The Future of IoT. *IEEE Internet of Things Magazine*, 28-33. doi:10.1109/IOTM.2018.1700021

Ploennigs, J., Cohn, J., Stanford-Clark, A., & IBM. (2018). The Future of IoT. *IEEE Internet of Things Magazine, 1*(1), 28-33. doi:10.1109/IOTM.2018.1700021

Popovic, T., Latinovic, N., Pešic, A., Zecˇevic, Z., Krstajic, B., & Djukanovic, S. (2017). Architecting an IoT-enabled platform for precision agriculture and ecological monitoring: A case study. *Computers and Electronics in Agriculture, 140*, 255–265. doi:https://doi.org/10.1016/j.compag.2017.06.008

Porcar-Castell, A., Garcia-Plazaola, J. I., Nichol, C. J., Kolari, P., Olascoaga, B., Kuusinen, N., . . . Nikinmaa, E. (2012). Physiology of the seasonal relationship between the photochemical reflectance index and photosynthetic light use efficiency. *Oecologia, 170*, 313-323. doi:10.1007/s00442-012-2317-9

Rojas, M. Z., Palomino, J., & Chávez, P. (2017). Development of a dedicated controller for a 5 MP CMOS sensor applied to a wireless image sensor network for large mammals monitoring. *2016 IEEE ANDESCON.* Arequipa, Peru: IEEE. doi:10.1109/ANDESCON.2016.7836234

Rouse, M. (2020, February). *internet of things (IoT).* Retrieved from Internet of things agenda: https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT

Rustia, D. J., & Lin, T.-T. (2017). An IoT-based Wireless Imaging and Sensor Node System for Remote Greenhouse Pest Monitoring . *CHEMICAL ENGINEERING TRANSACTIONS, 58* , 601-606. doi:10.3303/CET1758101

Ryu, Y., D.Baldocchi, D., Verfaillie, J., Ma, S., Falk, M., Ruiz-Mercado, I., . . . Sonnentag, O. (2010). Testing the performance of a novel spectral reflectance sensor, built with light emitting diodes (LEDs), to monitor ecosystem metabolism, structure and function. *Agricultural and Forest Meteorology, 150*(12), 1597-1606. doi:10.1016/j.agrformet.2010.08.009

Setiyono, T., Weiss, A., Specht, J., Cassman, K., & Dobermann, A. (2008). Leaf area index simulation in soybean grown under near-optimal conditions. *Field Crops Research, 108*, 82–92. doi:https://www.researchgate.net/deref/http%3A%2F%2Fdx.doi.org%2F10.1016%2Fj.fcr.2008.03.005

Sinha, N., Pujitha, K. E., & Alex, J. S. (2015, 1 22). Xively based sensing and monitoring system for IoT. *2015 International Conference on Computer Communication and Informatics (ICCCI -2015)* (pp. 1-6). Chennai,Tamil Nadu, India: IEEE. doi:10.1109/ICCCI.2015.7218144

Spectrum Technologies, I. (2011). *SpecWare Pro Disease Models Guide.* Spectrum Technologies, Inc. Retrieved from https://www.specmeters.com/assets/1/22/SpecWare_9_Disease_Models1.pdf

Statista. (2021, Nanuary 04). *statistics*. Retrieved from statista: https://www.statista.com/statistics/802690/worldwide-connected-devices-by-access-technology/#statisticContainer

Toldinas, J., Lozinskis, B., Baranauskas, E., & Dobrovolskis, A. (2019). MQTT Quality of Service versus Energy Consumption. *2019 23rd International Conference Electronics.* Palanga, Lithuania: IEEE. doi:10.1109/ELECTRONICS.2019.8765692

Tremblay, N., Wang, Z., Ma, B.-L., Belec, C., & Vigneault, P. (2008). A comparison of crop data measured by two commercial sensors for variable-rate nitrogen application. *Precision Agriculture*, 145-161. doi:10.1007/s11119-008-9080-2

Tzounis, A., Katsoulas, N., Bartzanas, T., & Kittas, C. (2017). Internet of Things in agriculture, recent advances and future challenges. *Biosystems Engineering, 164*, 31-48. doi:https://doi.org/10.1016/j.biosystemseng.2017.09.007

Vashi, S., Ram, J., Modi, J., Verma, S., & Prakash, C. (2017). Internet of Things (IoT): A vision, architectural elements, and security issues. *International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)* (pp. 492-496). Palladam, India: IEEE. doi:10.1109/I-SMAC.2017.8058399

Vasisht, D., Kapetanovic, Z., Won, J., Chandra, R., Kapoor, A., Sinha, S., . . . Stratman, S. (2017). FarmBeats: An IoT Platform for Data-Driven Agriculture. *14th {USENIX} Symposium on Networked Systems Design and Implementation* (pp. 515-529). Boston: USENIX Association. Retrieved from https://www.usenix.org/system/files/conference/nsdi17/nsdi17-vasisht.pdf

Vuran, M., Salam, A., Wong, R., & Irmak, S. (2018). Internet of underground things in precision agriculture: Architecture and technology aspects and technology aspects. *Ad Hoc Networks, 81*, 160–173. doi:https://doi.org/10.1016/j.adhoc.2018.07.017

APPENDIX

Two programs were used at MATLAB one is for real-time spectral signature display and the next is for real-time NDVI calculation. These two programs are presented below.

1. MATLAB Reflectance Calculation (Spectral Signature) and Graph Plotting

```
%Important – Be careful about the syntax errors you may face when copy and paste this code to the Thingspeak visualization platform.
%Developed by Nipuna Chamara for MS Research Project
%When implementing this program for each node first need to find the three channel IDs of the down looking device and the three channel IDs of the up looking senser, the channel ID parameter is different in each down looking sensor node. These three Channel IDs are unique for each device.
% Channel 1059365 1059539 and 1059542 contains data from the UNL Spider CAM site IoT device 10
% Channel ID to read data from device 01 included below
readChannelID1 = 1092483;
readChannelID2 = 1092484;
readChannelID3 = 1092485;
readChannelIDul1 = 1094383;
readChannelIDul2 = 1094386;
readChannelIDul3 = 1094390;
% Channel Read API Key
% Key between the " below:
readAPIKey1 = 'DEOJ5AZ5SZ9JZO4Q';
readAPIKey2 = '8G3GJ74Z2TGVCBHN';
readAPIKey3 = 'AFYYVGP9QAS7993Z';
readAPIKeyul1 = 'HUWO0G7QPOD4ITUH';
readAPIKeyul2 = 'SFVRS651YKD5DY63';
readAPIKeyul3 = 'Z5U3BXNCTJDTEJR1';
%Down looking Device spectral response obtainer
```

```
%Collect and average the past hour (60-minute period) down looking sensor node readings,
within each Channel there are 6 Fields
data1 =
thingSpeakRead(readChannelID1,'Fields',[1,2,3,4,5,6],'Numminutes',60,'OutputFormat','TimeTab
le', 'ReadKey',readAPIKey1)
dataHourly1 = retime(data1,'hourly','mean');

data2 =
thingSpeakRead(readChannelID2,'Fields',[1,2,3,4,5,6],'Numminutes',60,'OutputFormat','TimeTab
le', 'ReadKey',readAPIKey2)
dataHourly2 = retime(data2,'hourly','mean');

data3 =
thingSpeakRead(readChannelID3,'Fields',[1,2,3,4,5,6],'Numminutes',60,'OutputFormat','TimeTab
le', 'ReadKey',readAPIKey3)
dataHourly3 = retime(data3,'hourly','mean');

%Collect and average the past hour (60-minute period) up looking sensor node readings, within
each Channel there are 6 Fields

dataul1 =
thingSpeakRead(readChannelIDul1,'Fields',[1,2,3,4,5,6],'Numminutes',60,'OutputFormat','TimeT
able', 'ReadKey',readAPIKeyul1)
dataHourlyul1 = retime(dataul1,'hourly','mean');

dataul2 =
thingSpeakRead(readChannelIDul2,'Fields',[1,2,3,4,5,6],'Numminutes',60,'OutputFormat','TimeT
able', 'ReadKey',readAPIKeyul2)
dataHourlyul2 = retime(dataul2,'hourly','mean');

dataul3 =
thingSpeakRead(readChannelIDul3,'Fields',[1,2,3,4,5,6],'Numminutes',60,'OutputFormat','TimeT
able', 'ReadKey',readAPIKeyul3)
```

```
dataHourlyul3 = retime(dataul3,'hourly','mean');
data12 = synchronize(dataHourly1,
dataHourly2,dataHourly3,dataHourlyul1,dataHourlyul2,dataHourlyul3);
head(data12,1)
%Perform the reflectance calculation, Call the fields with assigned name (in this case the
wavelength)
%Refer the Equation 2.3 reflectance calculation in Chapter 2 to identify the equation.
%Reflectance coefficient was derived at the field calibration stage
Xul = [410 435 460 485 510 535 560 585 610 645 680 705 730 760 810 860 900 940];
Yul = [19.91*ans.A410nm/ans.AUL410nm 18.17*ans.B435nm/ans.BUL435nm
23.11*ans.C460nm/ans.CUL460nm 26.22*ans.D485nm/ans.DUL485nm
35.24*ans.E510nm/ans.EUL510nm 65.39*ans.F535nm/ans.FUL535nm
106.01*ans.G560Nm/ans.GUL560Nm 84.95*ans.H585Nm/ans.HUL585Nm
81.98*ans.R610nm/ans.RUL610nm 95.27*ans.I645nm/ans.IUL645nm
87.02*ans.S680nm/ans.SUL680nm 82.79*ans.J705nm/ans.JUL705nm
102.45*ans.T730nm/ans.TUL730nm 100.62*ans.U760nm/ans.UUL760nm
95.50*ans.V810nm/ans.VUL810nm 94.40*ans.W860nm/ans.WUL860nm
58.90*ans.K900nm/ans.KUL900nm 104.97*ans.L940nm/ans.LUL940nm];

%Plot the reflectance value (Spectral Signature)
plot(Xul,Yul);
xlabel('Electromagnetic Wavelength nm')
ylabel('Reflectance %');
title('Device 10 Reflectance');
```

2. MATLAB NDVI Calculation (Spectral Signature) and Graph Plotting

```
%%Important – Be careful about the syntax errors you may face when copy and paste this code to the ThingSpeak visualization platform.
%Developed by Nipuna Chamara for MS Research Project
%NDVI calculation
% When implementing this program for each node first need to find the three channel IDs of the down looking device and the three channel IDs of the up looking senser, the channel ID parameter is different in each down looking sensor node. These three Channel IDs are unique for each device.
% Channel 1094383 1094386 and 1094390 contains data from the UNL Spider CAM site IoT device 01
% Channel ID to read data from device 01
readChannelID1 = 1085083;
readChannelID2 = 1085084;
readChannelID3 = 1085085;

readChannelIDul1 = 1094383;
readChannelIDul2 = 1094386;
readChannelIDul3 = 1094390;

% Channel Read API Key is a must to read the data from the channel
% Key between the " below:
readAPIKey1 = 'IRNO8IPC5IR9FHG6';
readAPIKey2 = '7M54FQN72WRYJK40';
readAPIKey3 = 'CAGMSARQP4SJKCR6';

readAPIKeyul1 = 'HUWO0G7QPOD4ITUH';
readAPIKeyul2 = 'SFVRS651YKD5DY63';
readAPIKeyul3 = 'Z5U3BXNCTJDTEJR1';

%Down looking Device Channel data reads for the past hour and get the mean of those values
```

```
%Each channel has 6 fields to read

data1 =
thingSpeakRead(readChannelID1,'Fields',[1,2,3,4,5,6],'Numminutes',60,'OutputFormat','TimeTab
le', 'ReadKey',readAPIKey1)
dataHourly1 = retime(data1,'hourly','mean');

data2 =
thingSpeakRead(readChannelID2,'Fields',[1,2,3,4,5,6],'Numminutes',60,'OutputFormat','TimeTab
le', 'ReadKey',readAPIKey2)
dataHourly2 = retime(data2,'hourly','mean');

data3 =
thingSpeakRead(readChannelID3,'Fields',[1,2,3,4,5,6],'Numminutes',60,'OutputFormat','TimeTab
le', 'ReadKey',readAPIKey3)
dataHourly3 = retime(data3,'hourly','mean');

%Up looking Device spectral response collection for past hour and get the mean of them

dataul1 =
thingSpeakRead(readChannelIDul1,'Fields',[1,2,3,4,5,6],'Numminutes',60,'OutputFormat','TimeT
able', 'ReadKey',readAPIKeyul1)
dataHourlyul1 = retime(dataul1,'hourly','mean');

dataul2 =
thingSpeakRead(readChannelIDul2,'Fields',[1,2,3,4,5,6],'Numminutes',60,'OutputFormat','TimeT
able', 'ReadKey',readAPIKeyul2)
dataHourlyul2 = retime(dataul2,'hourly','mean');

dataul3 =
thingSpeakRead(readChannelIDul3,'Fields',[1,2,3,4,5,6],'Numminutes',60,'OutputFormat','TimeT
able', 'ReadKey',readAPIKeyul3)
dataHourlyul3 = retime(dataul3,'hourly','mean');
```

```
data12 =
synchronize(dataHourly1,dataHourly2,dataHourly3,dataHourlyul1,dataHourlyul2,dataHourlyul3)
;
head(data12,1)

%NDVI value calculation based on the Chapter 2 Equation 2.4
%First find the reflectance based on the selected wavelengths 680 nm and 860 nm
RRED = 87.02*ans.S680nm/ans.SUL680nm ;
RNIR = 94.40*ans.W860nm/ans.WUL860nm ;
%Use the reflectance on NDVI calculation
NDVI = (RNIR-RRED)/(RNIR+RRED);
NDVI = round(NDVI,2)
%Display the values on a ThingSpeak Channel dedicated to post NDVI values
display(NDVI,'NDVI');
thingSpeakWrite(1094453,'Fields',[4],'Values',{NDVI},'WriteKey','UDCU6KCMWK9A80S0')
```

3. ARDUINO SPECTRAL VALUE TRANSMISSION

```
/*
Developed by: Nipuna Chamara
Referred codes: https://github.com/sparkfun/SparkFun_AS7265x_Arduino_Library
How to use this code: Copy and paste on the Arduino IDE and compile, refer the wiring diagram to make the system before program the Arduino MKR1300
Date: April 01st, 2020

Functionality:
    1) Read the 18 channels of spectral sensor over I2C using the Spectral Triad
    2) Read the analog soil water content
    3) Save the data in the SD card
    4) Turn on and Off with delays the sensors connected to the Solar power managed
    5) Transmit the spectral sensor readings to the Things Speak Cloud via LPWAN

Hardware Connections:
Spectral sensors plug to Arduino MKR1300 to the I2C side of the AS7265x sensor
Open the serial monitor at 9600 baud to see the output
SD card
        SD card attached to SPI bus as follows:
        ** MOSI - pin 11
        ** MISO - pin 12
        ** CLK - pin 13
        ** CS - pin 4 (for MKRZero SD: SDCARD_SS_PIN)
Analog soil moisture Sensor: Connected to A0 pin MKR 1300
Battery voltage reading: Connected to A1 pin in MKR 1300
Solar Power Management Board Control: Connected to GPIO Pin 3 in Arduino MKR1300
```

Device ID: Device ID is the unique ID that represents this program included device, It could be set as a parameter that can be changed with
external input. But in this case, we did not include it because there are no such inputs (Text input by keyboard) for these devices.
*/

```
#include "SparkFun_AS7265X.h"
//Click here to get the library: http://librarymanager/All
#SparkFun_AS7265X
AS7265X sensor;
#include <SPI.h>
#include <SD.h>
#include <LoRa.h>

int soilmoisture = A0;    // select the input pin for the soilmoisture sensor
int batterylevel = A1;    // select the input pin for the battery voltage check
int powermanagement = 3;  // select the pin for the Solar power manager control
int sm = 0;               // variable to store the value coming from the sensor
int bl = 0;              // variable to store the value coming from the battery voltage level
const int chipSelect = 4; // ** CS - pin 4 (for MKRZero SD: SDCARD_SS_PIN)
int delay1=1000;          // This delay allows to set up reasonable time interval to give warm up time for sensors
int delay2=2000;          // This delay allows to set up reasonable gaps between different sensor activations
int delay3=6000;          // This delay decide the time interval between two messages sent by this device
int count=0;
int device_id=10009;     // ID of this End node, MQTT Channel Server settings in the outdoor LG02 unit Local Channel in/var/iot/channels has this ID

//Main loop stat here, check the sensor connection is the main use of this code, can include a warning code about sensor failures

void setup() {
          pinMode(powermanagement, OUTPUT);
          pinMode(5, OUTPUT);
          digitalWrite(5, HIGH);
          Wire.begin(60);
          Serial.begin(9600);

  while (!Serial) {
        //   ;
        // wait for serial port to connect. Needed for native USB port only
```

```
        //Uncomment this function when implement the sensor node, this is just to check
        the sensor node functionality for trouble shooting purpose
                }

Serial.print("Initializing SD card...");// see if the card is present and can be initialized:

 if (!SD.begin(chipSelect)) {
            Serial.println("Card failed, or not present");// don't do anything more:
            while (1);
                                    }
Serial.println("card initialized.");
digitalWrite(powermanagement, HIGH);

        delay(delay1); // see if the spectral sensor is present and can be initialized:

 if(sensor.begin() == false)
       {
                Serial. println("Sensor does not appear to be connected. Please check the
                wiring. Freezing...");
                while(1);
       }

Serial.println("A,B,C,D,E,F,G,H,I,J,K,L,R,S,T,U,V,W");
         delay(delay1);
digitalWrite(powermanagement, LOW);
         delay(delay1);
         }

void loop() {

digitalWrite(powermanagement, HIGH); // Turn on the sub systems including sensors
and image sensor board
         delay(delay2);            // Give reasonable time to warm up sensors and devices
  // read the value from the soil moisture sensor:
  float sm = analogRead(soilmoisture);
  //Serial.begin(115200);
    Serial.print(sm*5/1024);
    Serial.println("Voltage");
    Serial.print(((sm*5/1024)*0.494)-0.554);// Soil moisture calibration equation
recommended by the OEM
    sm=((sm*5/1024)*0.494)-0.554;
    Serial.println("v/v");
    Serial.println(sm);
    delay(delay1); // read the value from the battery sensor:
```

```
   float bl = analogRead(batterylevel); //Serial.begin(115200);
   bl=bl*5/1024;
   Serial.println(bl);
   delay(delay1); // see if the spectral sensor is present and can be initialized:

   if(sensor.begin() == false)
     {
Serial. println("Sensor does not appear to be connected. Please check the wiring.
Freezing...");
       while(1);
     }
  Serial.println("A,B,C,D,E,F,G,H,I,J,K,L,R,S,T,U,V,W");

  sensor.takeMeasurementsWithBulb(); //This is a hard wait while all 18 channels are
measured

            double A = sensor.getCalibratedA();
            double B = sensor.getCalibratedB();
            double C = sensor.getCalibratedC();
            double D = sensor.getCalibratedD();
            double E = sensor.getCalibratedE();
            double F = sensor.getCalibratedF();
            double G = sensor.getCalibratedG();
            double H = sensor.getCalibratedH();
            double I = sensor.getCalibratedI();
            double J = sensor.getCalibratedJ();
            double K = sensor.getCalibratedK();
            double L = sensor.getCalibratedL();
            double R = sensor.getCalibratedR();
            double S = sensor.getCalibratedS();
            double T = sensor.getCalibratedT();
            double U = sensor.getCalibratedU();
            double V = sensor.getCalibratedV();
            double W = sensor.getCalibratedW();

delay(delay2);
  String dataString1 = ""; //string variable to store sensor A,B,C,D,E and F
  String dataString2 = ""; //string variable to store sensor G,H,I,J,K and L
  String dataString3 = ""; //string variable to store sensor R,S,T,U,V and W // read three
sensors and append to the string:
  sensor.takeMeasurements();
    dataString1 = String(A)+" "+String(B)+" "+String(C)+" "+String(D)+" "+String(E)+"
"+String(F);
```

```
   dataString2 = String(G)+" "+String(H)+" "+String(I)+" "+String(J)+" "+String(K)+"
"+String(L);
   dataString3 = String(R)+" "+String(S)+" "+String(T)+" "+String(U)+" "+String(V)+"
"+String(W); // see if the card is present and can be initialized:
     if (!SD.begin(chipSelect)) {
       Serial.println("Card failed, or not present"); // don't do anything more:
       while (1);
     }
     Serial.println("card initialized.");
     delay(delay1);  // open the file. note that only one file can be open at a time,
                     // so you have to close this one before opening another.
 File dataFile1 = SD.open("datalog1.txt", FILE_WRITE);// if the file is available, write to
it A,B,C,D,E,F:
  if (dataFile1) {
    dataFile1.println(dataString1);
    dataFile1.close();// print to the serial port too:
    delay(delay1);
    Serial.println(dataString1);
     } // if the file is available, write to it G,H,I,J,K,L:

 File dataFile2 = SD.open("datalog2.txt", FILE_WRITE);
    // if the file is available, write to it:
  if (dataFile2) {
    dataFile2.println(dataString2);
    dataFile2.close();    // print to the serial port too:
    delay(delay1);
    Serial.println(dataString2);
     }// if the file is available, write to it R,S,T,U,V,W:

 File dataFile3 = SD.open("datalog3.txt", FILE_WRITE);
     // if the file is available, write to it:
  if (dataFile3) {
    dataFile3.println(dataString3);
    dataFile3.close();
    // print to the serial port too:
    delay(delay1);
    Serial.println(dataString3);
     }    // if the file isn't open, pop up an error:
  else {
    Serial.println("error opening datalog.txt");
  }
  delay(delay1);
  Serial.print("Sending packet: ");
  Serial.println(count);
```

```
  digitalWrite(5, LOW);
  delay(delay1); // see if the LoRa hardware is present and can be initialized:
     Serial.println("LoRa Sender");
     if (!LoRa.begin(915E6)) {
       Serial.println("Starting LoRa failed!");
       while (1);
     }
     LoRa.setSyncWord(0x34);

delay(delay2);  // compose and send packet
    int AL = A;
    int BL = B;
    int CL = C;
    int DL = D;
    int EL = E;
    int FL = F;
    int GL = G;
    int HL = H;
    int IL = I;
    int JL = J;
    int KL = K;
    int LL = L;
    int RL = R;
    int SL = S;
    int TL = T;
    int UL = U;
    int VL = V;
    int WL = W;

// Below 22 repeated codes are to send 22 different sensor readings to Gateway, It is
possible to put this stuff in a loop.
// with proper variables
digitalWrite(powermanagement, LOW); // Turn off all the subsystems to save the battery
energy
  LoRa.beginPacket();
  LoRa.print("<");
  LoRa.print(device_id);
  LoRa.print(">field1=");
  LoRa.print(AL); // LoRa.print(counter);
  LoRa.endPacket();
  count++;

delay(delay3);
LoRa.beginPacket();
```

```
  LoRa.print("<");
  LoRa.print(device_id);
    LoRa.print(">field2=");
    LoRa.print(BL); // LoRa.print(counter);
  LoRa.endPacket();
  count++;

 delay(delay3);
LoRa.beginPacket();
  LoRa.print("<");
  LoRa.print(device_id);
    LoRa.print(">field3=");
    LoRa.print(CL); // LoRa.print(counter);
  LoRa.endPacket();
  count++;

 delay(delay3);
LoRa.beginPacket();
  LoRa.print("<");
  LoRa.print(device_id);
    LoRa.print(">field4=");
    LoRa.print(DL); // LoRa.print(counter);
  LoRa.endPacket();
  count++;

 delay(delay3);
LoRa.beginPacket();
  LoRa.print("<");
  LoRa.print(device_id);
    LoRa.print(">field5=");
    LoRa.print(EL); // LoRa.print(counter);
  LoRa.endPacket();
  count++;
  delay(delay3);
LoRa.beginPacket();
  LoRa.print("<");
  LoRa.print(device_id);
    LoRa.print(">field6=");
    LoRa.print(FL); // LoRa.print(counter);
  LoRa.endPacket();
  count++;
//soil moisture
  delay(delay3);
LoRa.beginPacket();
```

```
  LoRa.print("<");
  LoRa.print(device_id);
    LoRa.print(">field7=");
    LoRa.print(sm);
 // LoRa.print(counter);
  LoRa.endPacket();
  count++;
//battery level
  delay(delay3);
LoRa.beginPacket();
  LoRa.print("<");
  LoRa.print(device_id);
    LoRa.print(">field8=");
    LoRa.print(bl);
 // LoRa.print(counter);
  LoRa.endPacket();
  count++;

//Device iD 10009
    delay(delay3);
LoRa.beginPacket();
  LoRa.print("<");
  LoRa.print(device_id+1);
    LoRa.print(">field1=");
    LoRa.print(GL);
 // LoRa.print(counter);
  LoRa.endPacket();
  count++;

    delay(delay3);
LoRa.beginPacket();
  LoRa.print("<");
  LoRa.print(device_id+1);
    LoRa.print(">field2=");
    LoRa.print(HL);
 // LoRa.print(counter);
  LoRa.endPacket();
  count++;

    delay(delay3);
LoRa.beginPacket();
  LoRa.print("<");
  LoRa.print(device_id+1);
    LoRa.print(">field3=");
```

```
    LoRa.print(IL);
 // LoRa.print(counter);
  LoRa.endPacket();
  count++;

    delay(delay3);
LoRa.beginPacket();
  LoRa.print("<");
  LoRa.print(device_id+1);
    LoRa.print(">field4=");
    LoRa.print(JL);
 // LoRa.print(counter);
  LoRa.endPacket();
  count++;

     delay(delay3);
LoRa.beginPacket();
  LoRa.print("<");
  LoRa.print(device_id+1);
    LoRa.print(">field5=");
    LoRa.print(KL);
 // LoRa.print(counter);
  LoRa.endPacket();
  count++;

    delay(delay3);
LoRa.beginPacket();
  LoRa.print("<");
  LoRa.print(device_id+1);
    LoRa.print(">field6=");
    LoRa.print(LL);
 // LoRa.print(counter);
  LoRa.endPacket();
  count++;

  //Device iD 10010
    delay(delay3);
LoRa.beginPacket();
  LoRa.print("<");
  LoRa.print(device_id+2);
    LoRa.print(">field1=");
    LoRa.print(RL);
 // LoRa.print(counter);
  LoRa.endPacket();
```

```
  count++;

    delay(delay3);
LoRa.beginPacket();
  LoRa.print("<");
  LoRa.print(device_id+2);
    LoRa.print(">field2=");
    LoRa.print(SL);
 // LoRa.print(counter);
  LoRa.endPacket();
  count++;

    delay(delay3);
LoRa.beginPacket();
  LoRa.print("<");
  LoRa.print(device_id+2);
    LoRa.print(">field3=");
    LoRa.print(TL);
 // LoRa.print(counter);
  LoRa.endPacket();
  count++;

    delay(delay3);
LoRa.beginPacket();
  LoRa.print("<");
  LoRa.print(device_id+2);
    LoRa.print(">field4=");
    LoRa.print(UL);
 // LoRa.print(counter);
  LoRa.endPacket();
  count++;

    delay(delay3);
LoRa.beginPacket();
  LoRa.print("<");
  LoRa.print(device_id+2);
    LoRa.print(">field5=");
    LoRa.print(VL);
 // LoRa.print(counter);
  LoRa.endPacket();
  count++;

    delay(delay3);
LoRa.beginPacket();
```

```
  LoRa.print("<");
  LoRa.print(device_id+2);
    LoRa.print(">field6=");
    LoRa.print(WL);
 // LoRa.print(counter);
  LoRa.endPacket();
  count++;

  delay(delay3);
    digitalWrite(5, HIGH);
  delay(delay3);
}
```