**Final Report**

Ingram Readymix Secured Network

**Project Manager: Ahmed Al Qaysi**

**Andres oliva, Nicholas Holleman**

**Texas State University**

**Ingram School of Engineering**

**INGRAM READY MIX**

**TRANSPORTATION WAY**

**SAN MARCOS, TX 78666**

**Date**

**December 1, 2019**





**Test Plan Revision History**:

# Overview

## Executive Summary

By Andres Oliva:

The Ingram Readymix Secured Network is a project that focuses on creating a power management system for sustainability, a blockchain data ledger providing security and a graphical user interface (GUI) for data visualization on an existing system. These features were designed and created by a team of 3 electrical engineers. Ahmed Al Qaysi (project manager), with a discipline in computer engineering, Nicholas Holleman and Andres Oliva, with disciplines in micro and nano technology.

This project was sponsored by the Ingram Readymix cement company. The project is intended to create a sustainable system that will allow remote access and security to the retention pond located outside the office. Throughout the design and implementation process, the team was successful in reaching desired goals for the project. The power management showed success when taking power measurements of the old system and comparing them to the new upgraded system. Power management was implemented to extend the battery life of the outdoor system for a minimum of three days. In accordance to the TCEQ permit a data ledger was created to hold a minimum three years-worth of logs and the ability to verify data is not tampered with through the implementation of blockchain. Secured communication between the office and the retention pond was enhanced with a layer of encryption applied. The graphical user interface (GUI) was implemented to visualize the pond status. The GUI is required to demonstrate current and past status of the pond water levels, as well as the power status of the system.

The purpose of this project was intended to reduce human effort at the Ingram Readymix facility. Not only is this project valuable to the students, but as well as the Ingram facility. This will be beneficial to anyone who works at the concrete plant to protect the business from possible fines.

## Abstract

By Ahmed Al Qaysi:

Ingram Readymix is a cement plant that contains a water retention pond on site. Water collected at the pond consists of water used at the facility as well as rainfall water. Ingram must comply with the Texas Commission of Environmental Quality (TCEQ) permit requirements, which requires the company to collect a water sample within 30 minutes of water leaving through a storm drain. Our project will augment hardware at the Ingram concrete remote retention pond. This is done with supplementary components to monitor a power system. In addition, software is completely replaced and developed through python to create a GUI, network encryption, power management and a blockchain ledger for data.

Contents

[1 Overview 2](#_Toc509991144)

[1.1 Executive Summary 2](#_Toc509991145)

[1.2 Abstract 2](#_Toc509991146)

[2 List of Figures 4](#_Toc509991147)

[3 List of Tables 4](#_Toc509991148)

[4 Problem Description 4](#_Toc509991149)

[5 Progress Towards A Solution 7](#_Toc509991150)

[5.1 Design Decisions 7](#_Toc509991151)

[5.2 Design Approach 10](#_Toc509991152)

[5.3 Project Approach 16](#_Toc509991153)

[5.4 Engineering Standards 17](#_Toc509991154)

[5.5 Progress Towards Goals 17](#_Toc509991155)

[5.6 Verification 18](#_Toc509991156)

[5.7 Characterization Results 29](#_Toc509991157)

[5.8 Deficiencies 33](#_Toc509991158)

[5.9 Iterations and Redefinitions 34](#_Toc509991159)

[6 Constraints 34](#_Toc509991160)

[6.1 Budgetary 34](#_Toc509991161)

[6.2 Design Feasibility 35](#_Toc509991162)

[6.3 Manufacturability 35](#_Toc509991163)

[6.4 Maintainability 35](#_Toc509991164)

[6.5 Environmental 35](#_Toc509991165)

[6.6 Health and Safety 36](#_Toc509991166)

[6.7 Social 36](#_Toc509991167)

[7 Budgets 36](#_Toc509991168)

[8 Work Schedule 37](#_Toc509991169)

[9 Personnel Interactions 39](#_Toc509991170)

[9.1 Teamwork 39](#_Toc509991171)

[9.2 Mentorship 39](#_Toc509991172)

[10 Ethics 40](#_Toc509991173)

[11 Summary & Conclusions 40](#_Toc509991174)

[12 Discussion 40](#_Toc509991175)

[12.1 Academic Preparation 41](#_Toc509991176)

[12.2 Lessons Learned 41](#_Toc509991177)

[12.3 Soft Skills 42](#_Toc509991178)

[12.4 Schedule Deviations 42](#_Toc509991179)

[12.5 Staffing 42](#_Toc509991180)

[12.6 Final Observations 43](#_Toc509991181)

[13 Acknowledgments 43](#_Toc509991182)

[14 References 43](#_Toc509991183)

# List of Figures

Figure 1: Block diagram of overall system

Figure 2: Hardware Block Diagram of overall system

Figure 3: Micro-controller logic flow chart.

Figure 4: Microprocessor communication logic flow chart.

Figure 5: Pond communication class UML diagram.

Figure 6: Microprocessor power management logic diagram.

Figure 7: UML Object: block chain.

Figure 8: Office logic flow chart.

Figure 9: Office Communication UML class diagram.

Figure 10: Graphical User Activity Diagram.

# List of Tables

Table 1: Deliverable Specification.

Table 2: Applicable standards.

Table 3: Status of goals.

Table 4: Test cases.

Table 5: Results of test cases.

Table 6: Budget proposed versus actual spent.

Table 7: Work schedule proposed versus actual schedule.

Table 8: Academic preparation table.

# Problem Description

By Ahmed Al Qaysi:

Ingram Readymix is a cement plant that contains a water retention pond on the edge of the facility. The water collected at the pond holds water that is used at the facility to wash off cement mixture and collects rainfall water. Ingram Readymix must comply with the Texas Commission of Environmental Quality (TCEQ) permit requirements, which requires the company to collect a water sample within a time frame of 30 minutes when water leaves through a storm drain during an overflow event.

The objective of the project is to augment Ingram’s current system which consists of a battery, a solar panel, two antennas, a water sensor and an LED strip that signals a rise in the water levels at the pond. Currently, the system is vulnerable to third party attacks and lacks a self-sustainable power management system to provide power; a secure data ledger for data storage, and an effective GUI for system status and visualization purposes. Due to these deficiencies, our team goals for this project will consist of adding such features.

|  |  |
| --- | --- |
| **Features** | **Performance Targets** |
| **Graphical User Interface** | |
| Display | 7” Touchscreen with graphical user interface that can guide user through Alerts and battery health, and current/past water levels and replaces LED strip |
| Battery health | Track, monitor and forecast battery life based on solar panel, battery and battery control sensor. |
| **Diagnostic** | |
| Antenna detection | Antenna detection will verify both antennas, power when need. |
| Network connection | Check to see if the other pi is currently available on the network. |
| Solar panel voltage sensor | This sensor checks the solar panel voltage and will be used to determine if the panel can send current to the battery controller. |
| Solar panel current sensor | Solar current sensor will state if the battery control unit is accepting current from the solar panel to the charge battery. |
| Battery control Sensor | This sensor will determine if the system is causing an overload and confirm if another sensor is reasonable. |
| Battery voltage sensor | The battery voltage sensor will help detect the capacity of the battery being used for power. |
| Battery current sensor | The battery current sensor will calculate the current being drawn from battery being used for power. |
| Communication | |
| Client/Server application | Both the Pi computer in the office, as well as the Pi computer outside will communicate via TCP connection. |
| Encryption | Application level encryption with average transmit efficiency of 50% or greater. |
| Power management | |
| Battery life | Double battery life via power device when they are need. |
| Protection | Prevent system from powering microprocessor if voltage is too high or low. |
| Data Ledger | |
| Storage | Store atlest 3 years’ worth of sensor readings. |
| Integrity | Ability to detect alterations within ledger. |

*Table 1: Deliverable Specification.*

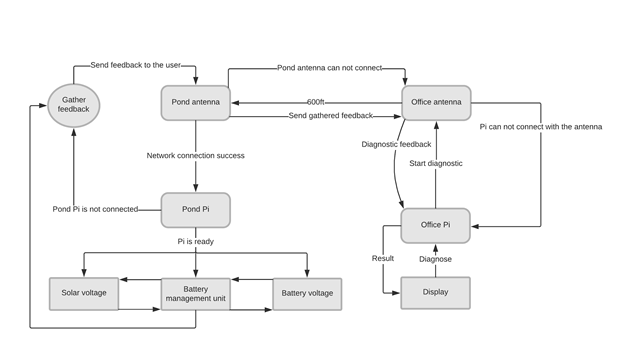


Figure 1: Block diagram of the overall system.

# Progress Towards A Solution

## Design Decisions

By Nicholas Holleman:

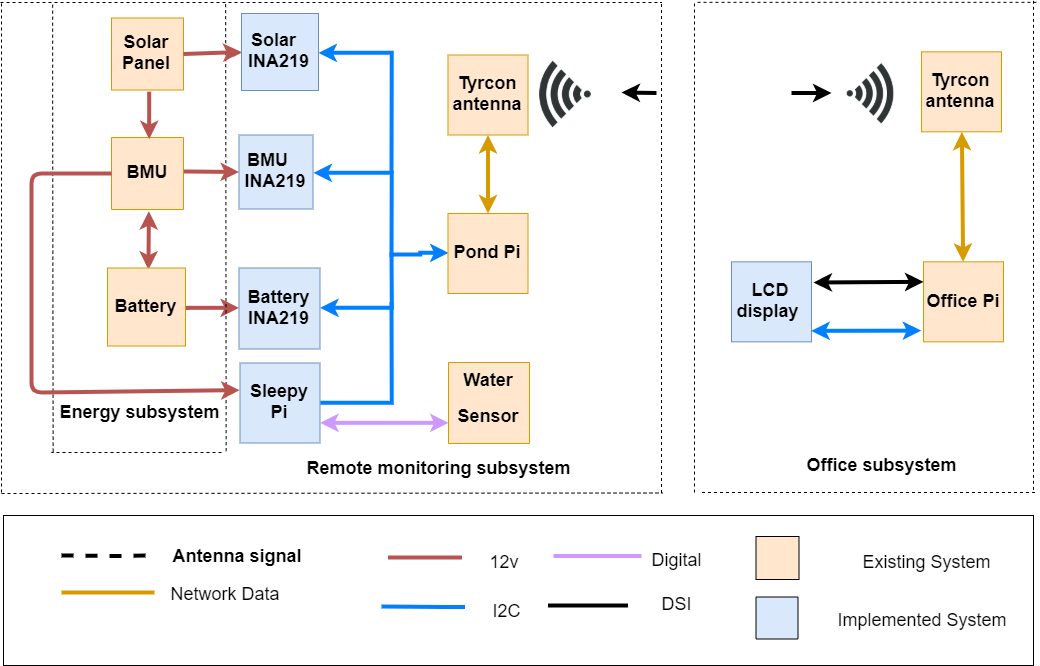
****

Figure 2: Hardware Block diagram of the overall system.

The design decisions made for this project were chosen based on the requirements established. The yellow boxes in Figure 2 represent hardware components that were already existent at the facility. The hardware components that our team added consist of three current and voltage sensors, two microprocessors, and a microcontroller power management unit. These hardware components are shown as blue boxes in Figure 2, which is a hardware block diagram of the overall system.

The team approached the project assuming the system was fully functional. This meant that Ingram Readymix was requesting to upgrade the system to include new features. This approach would give the team access to hardware that was previously implemented at the Ingram Readymix onsite retention pond. This approach allowed us to keep a small budget.

One of the major features added to the system was the ability to monitor the battery status. There were two different ways to approach this. One of those was to find a new battery management unit (BMU) and the second was to add sensors that could do this. A new BMU would have been preferable as it would provide pre-built functionality versus coming up with something new. The sensor option was chosen because the cost was only a fraction of a new BMU, and this included communication capabilities.

With power management as a major feature, this meant that Ingram Readymix was requesting to use an entirely new microprocessor or implement a micro-controller that could turn the microprocessor on and off. A major issue with replacing the microprocessor dealt with the cost and functionality. In terms of functionality, a new microprocessor would require the ability to save information via a micro SD card, network communication, a real time clock (RTC), and wake option on either network signal or general pin in/out mode. The microprocessor would also need RAM and enough processing speed for encryption. Before taking encryption into accountability, each of the other features required for the system to function would require their own dedicated hardware add-on for the new micro-processor; this is approximately the same cost as the micro-controller that was chosen for this portion of the project. This does not account for the cost of the micro-processor itself. Keeping this in mind, different microprocessors were not readily available with encryption libraries as well as the ability options to wake on GPIO pins or LAN.

The micro-controller chosen for the project was reasonably inexpensive. This device has a lower power consumption while running and generates a low power state itself. It can be directly mounted on the current micro-processor at Ingram ReadyMix facility. The micro-controller can read from all sensors and manage its own special voltage rail to power all sensors giving the ability to power them down as well or reset them when required.

The system prior contained an LED strip that would light up vertically to indicate the water levels of the retention pond. The choice was either leave it as is, add to it or replace. The team chose to replace the LED strip with a GUI to help provide feedback and the status on the functionality of the system. The 7” LCD screen allows the use of different software features such as battery level status, solar power generation, a rise in water levels and electrical currents.

Due to the nature of the remote location, communication becomes a major issue. The system was original designed with Wi-Fi Protected Access 2 (WPA2) encryption. Previously, there was no layer level of encryption or any other type of protection. With nothing present a transmission control protocol was chosen along with the application of layer encryption. The application layer encryption that was chosen for the project is RSA. This is an asymmetrical encryption along with TCP connection for its connection-oriented design. Each of these features were chosen for certain qualities. The TCP connection was chosen due to its simplicity, packet order, the request of lost packets in chain and the ability to accept bit/strings.

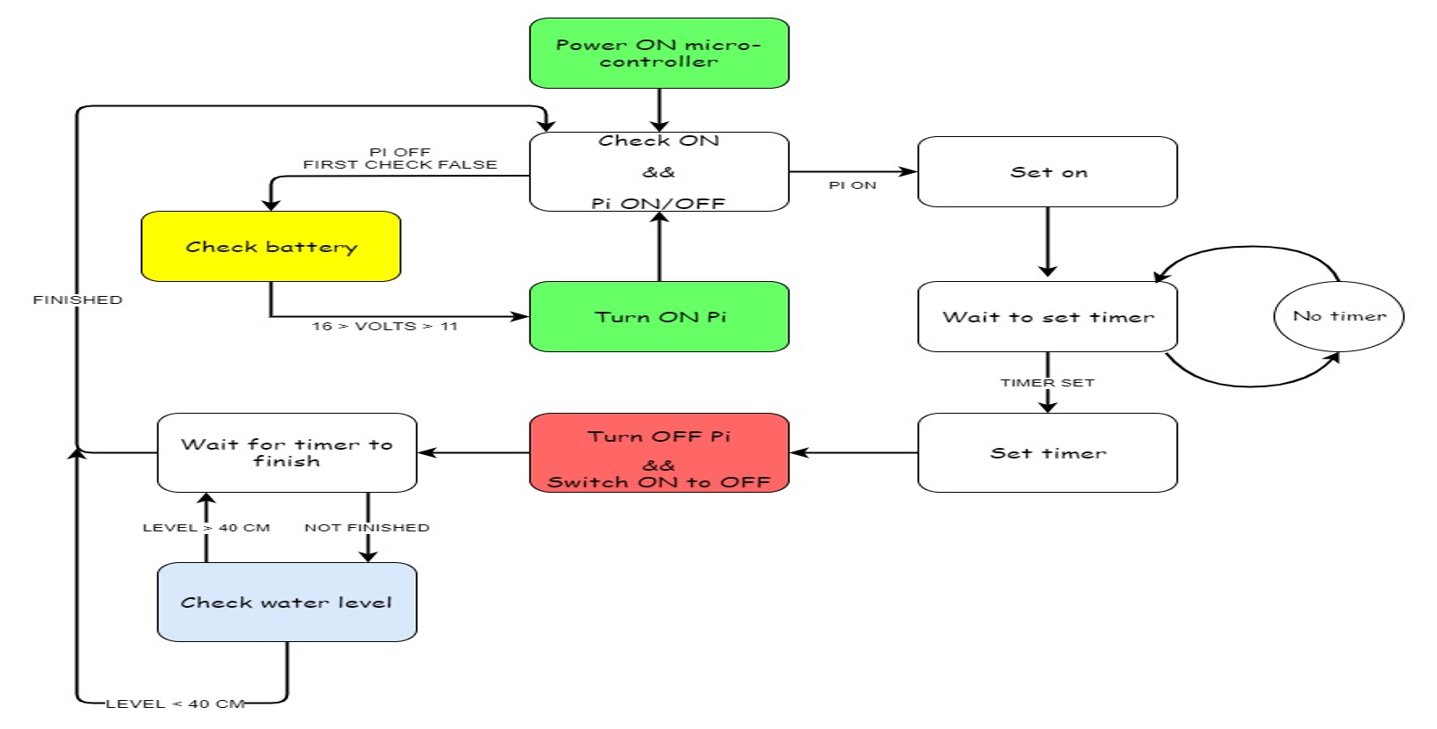
AES encryption was not chosen because it is less secure than what was desired. It means that each direction is encrypted with the same key; if one key got compromised, the entire system would then be compromised. RSA encryption encrypts both directions differently using private and public keys. This gave an added benefit if anyone was listening to the data stream, they would be required to understand each side as each side has a different key.

There are multiple ways to design a ledger, each with different complications. With integrity being important, several alternatives were discarded. One alternative with the ability to have multiple copies at multiple locations. Another method of a ledger could have been to use cyclic redundancy check (CRC). The issue without multiple copies of the checksum being distributed the CRC of the file could be altered and the file itself could then become altered. This led to the idea of using new technology such as blockchain. Blockchain was chosen as a structure, this means each block locks the previous block. This gives the system the ability to detect if any sensor, time, source, or alert message has been altered.

## Design Approach

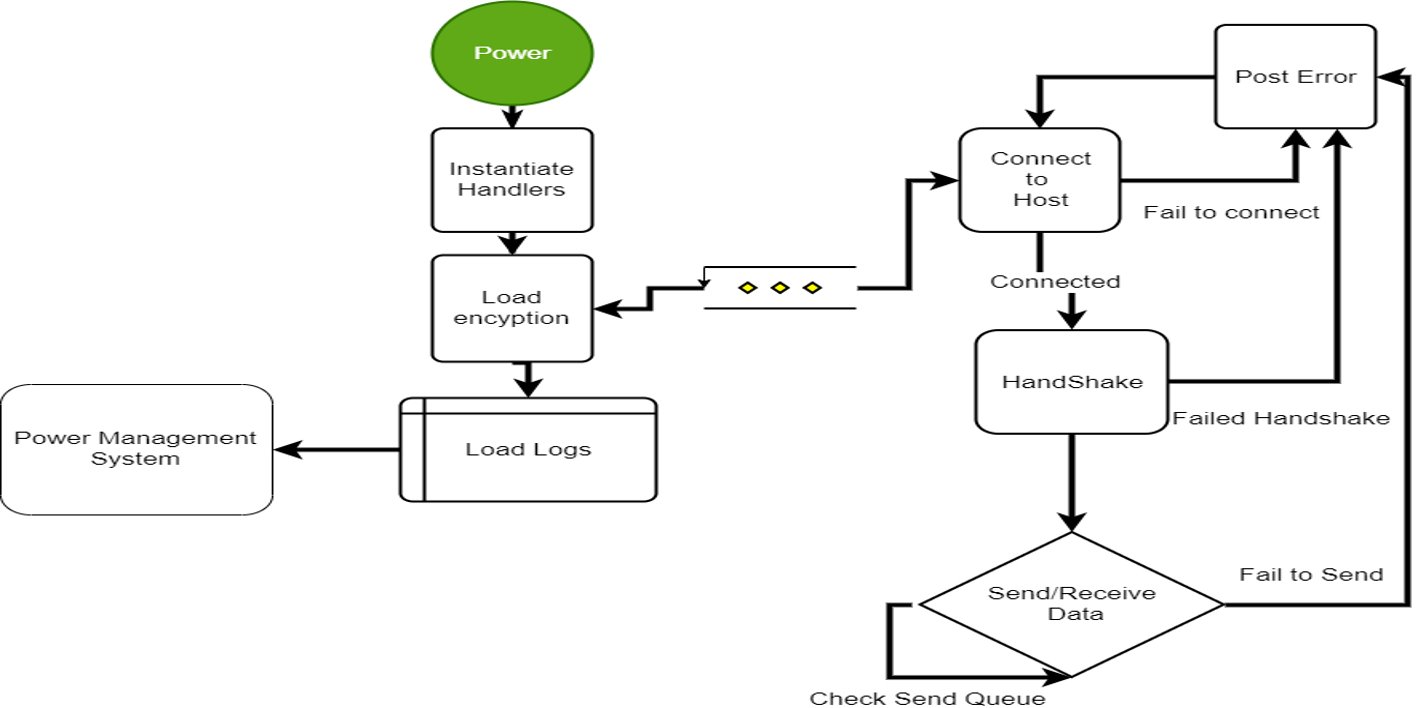
By Nicholas Holleman:

Majority of each system feature was designed firstly by deciding what was required out of the system and what dependency each system had to one another. Most features were designed using logic flow charts, block diagrams, universal modeling language (UML) object diagrams and UML class diagrams. Each system was also designed independently of each other and combined at a logical stop or starting point. The software used to create the diagrams was Draw.io, which is a website and a standalone diagram creation software that has pre-built symbols for different styles. The micro-controller logic was the last design, but it is also the first logic the water retention pond sees.



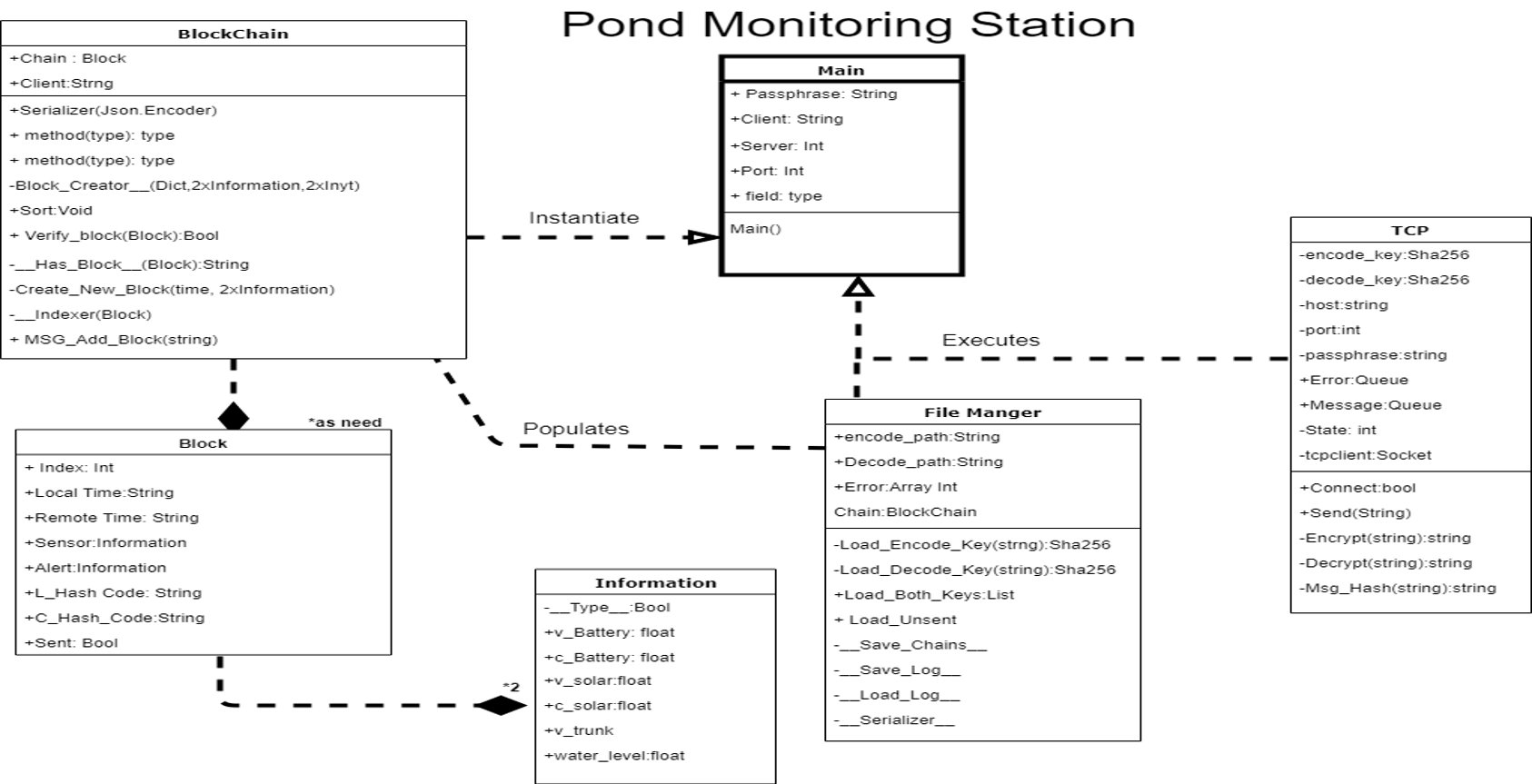
*Figure 3: Micro-controller logic flow chart.*

Figure 3 shows a diagram of the simplest logic on the micro-controller. Some of the blocks within this diagram have additional logic for the system to fulfill the required logic within the diagram. Due to the complicated system of two state machines having to talk to each other, the micro-processor logic follow chart had to be broken up into major sections. This was broken up into the communication and power management side.



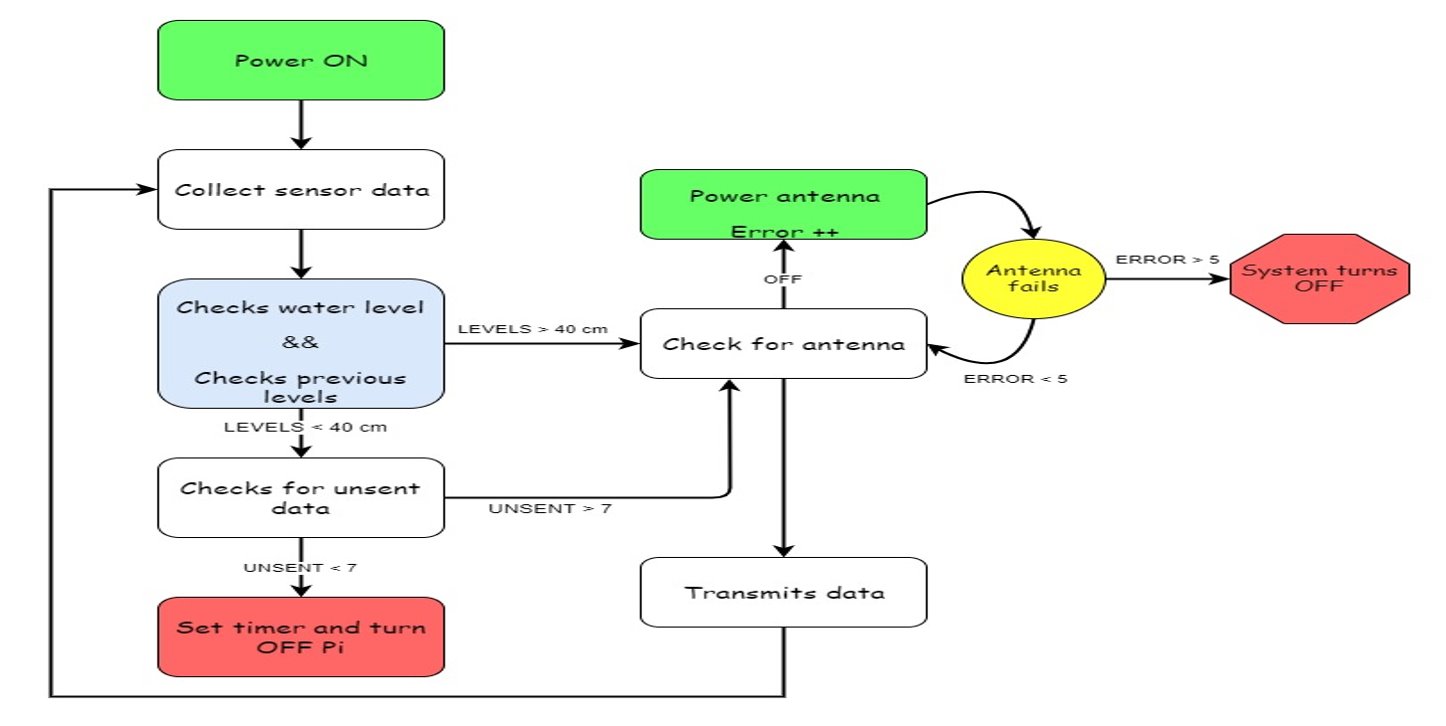
*Figure 4: Microprocessor communication logic flow chart.*

Figure 4 shows the simplified logic diagram of the basic communication system and loading of the ledger that the system requires to function. Within the handshake is a two-party verification system. The microprocessor sends a passphrase encrypted with the server’s public key and awaits a transmission back consisting of a passphrase encrypted with its personal public encryption key. If this handshake fails, the office and the pond will not communicate, and the data will be stored locally until fixed.



*Figure 5: Pond communication class UML diagram.*

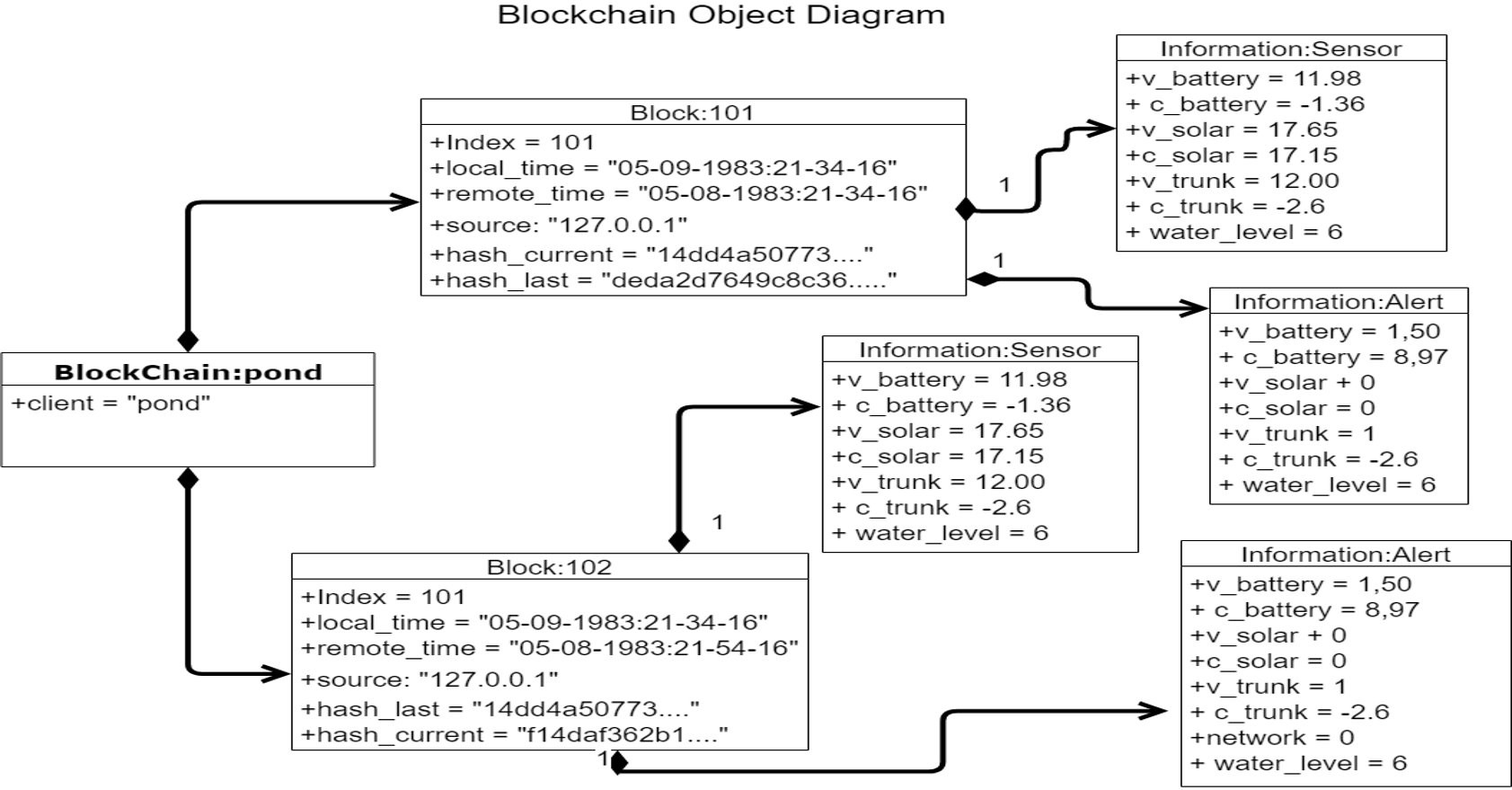
The diagram in Figure 5 was created before the classes or functions of the pond/water retention program; it was also updated for any missing functions as the project moved along. One of the features for the communication side of the project came out of the creation of this diagram. The system not only uses encryption and blockchain to verify the ledger, but it also uses a small isolated chain each time it connects to verify the contents of the messages sent to the office. This allows the office another chance to verify that the message is authentic, providing a way to respond to the pond and to inform that a block was received and not to send it again.



*Figure 6: Microprocessor power management logic diagram.*

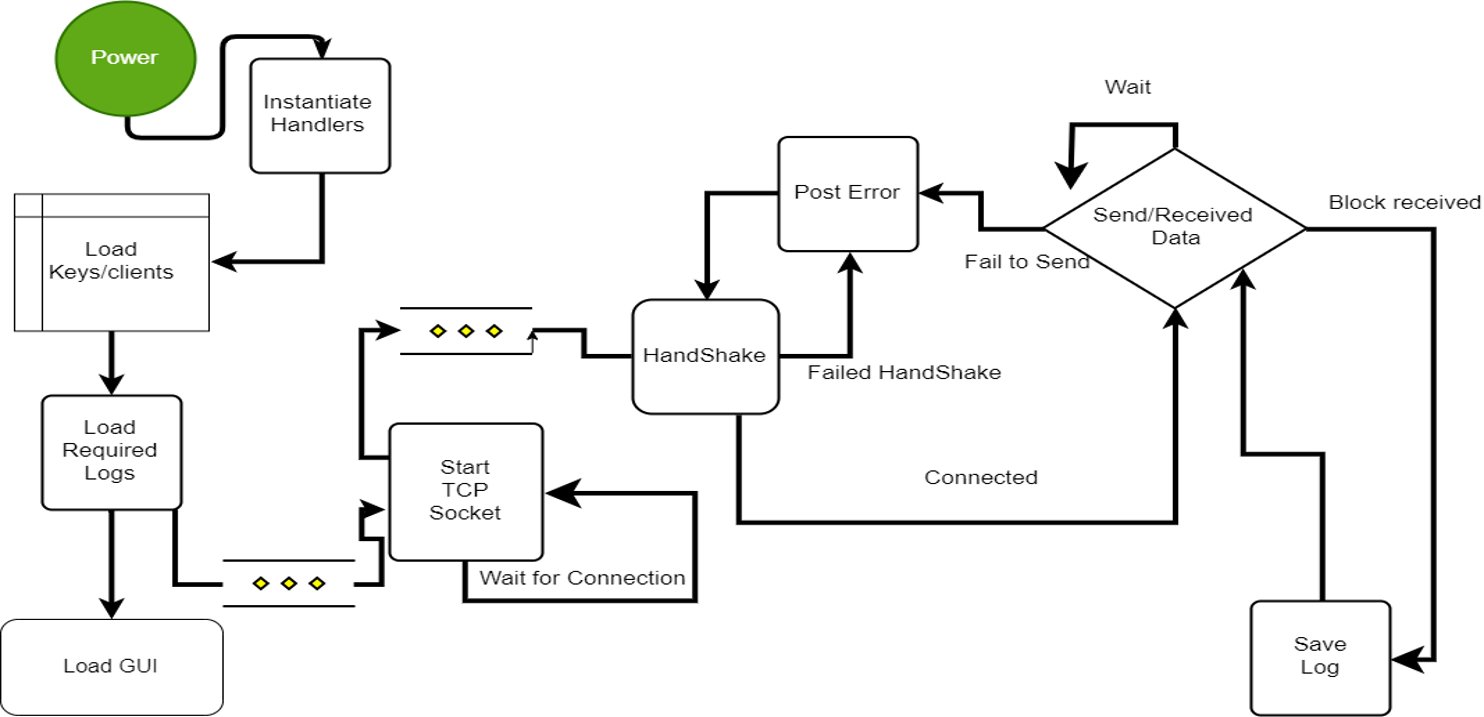
Figure 6 shows the logic diagram of the microprocessor after the logs are loaded. The logic and mode of operation was designed separately from the communication operation. The logic in this diagram comes directly after the load logs in Figure 4. The green box labeled power antenna means the system powers a relay that gives power to the system WI-FI Access point. The box that is labeled set timer is talking to the micro-controller to tell it how long to set a timer for and please turn me off via software commands.

The next portion of the system designed was the ledger. The ledger is the same on the water retention site and in the office. This allowed the system to power down without having to transmit each time it takes a power reading and save power.



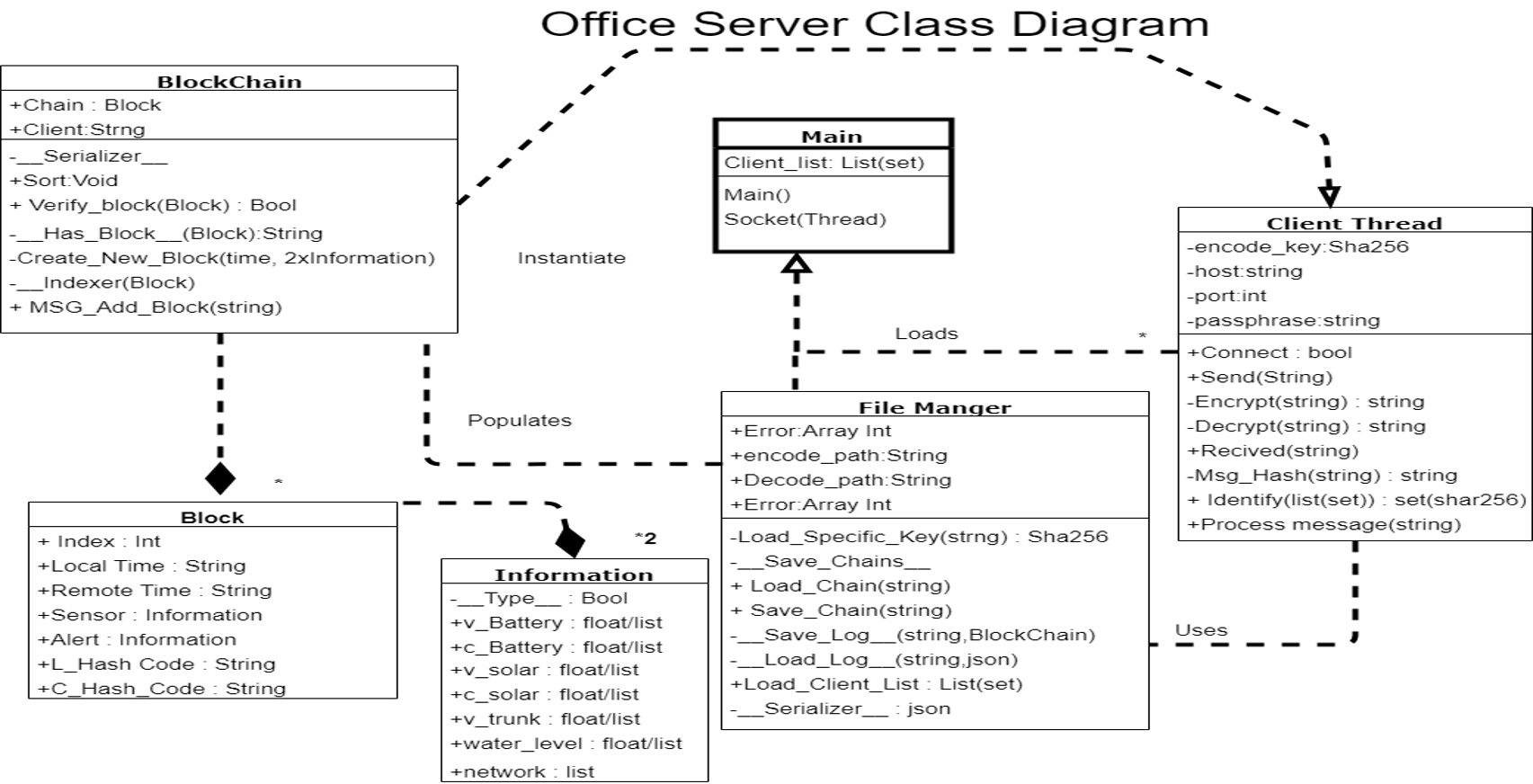
*Figure 7: UML Object: block chain.*

Figure 7 is a realized non-serialized representation of ledger. This diagram was designed so it was clear what data needs to be readily available to the person creating the GUI, also so that no required data from the log would be missed in either serial or non-serial form.



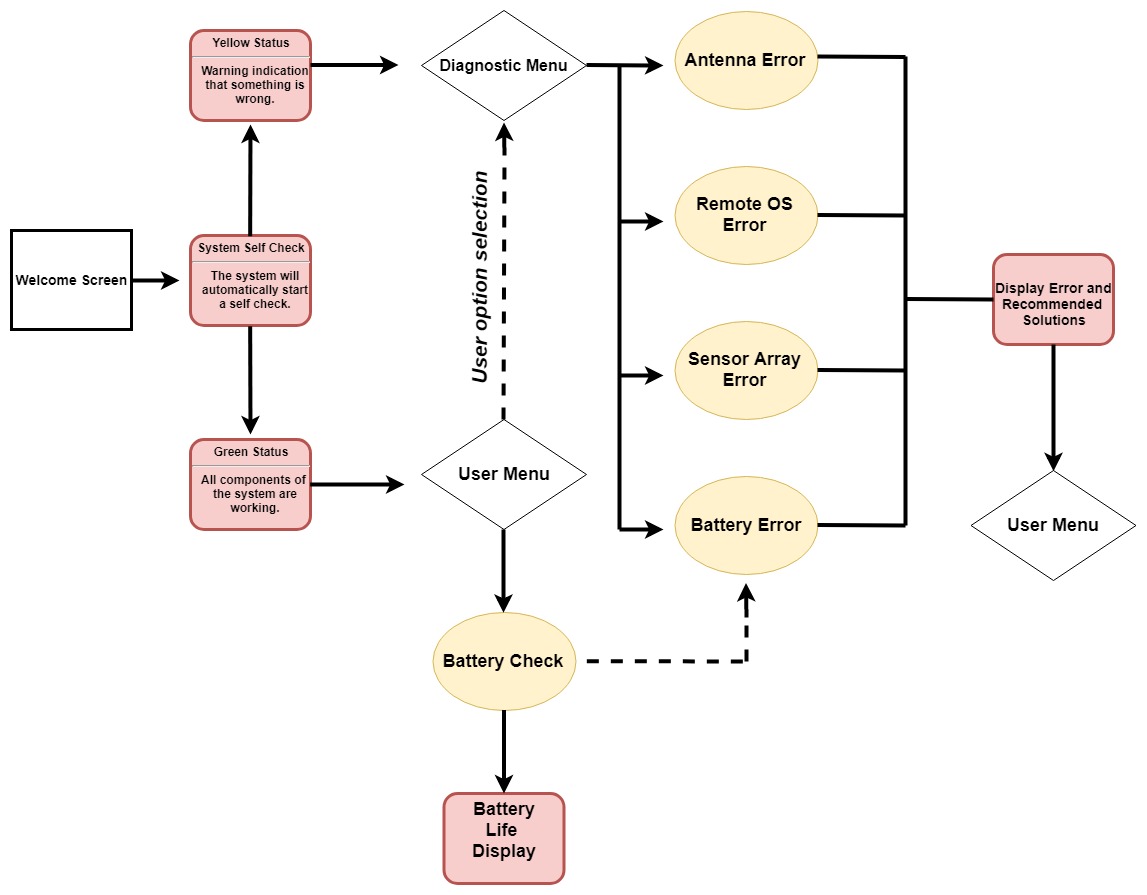
*Figure 8: Office logic flow chart.*

Figure 7 was created out of redundancy. The diagram explains logic that as almost identical to the pond, with the small difference of it is to load the GUI, after the logs are loaded. After its creation, a UML class diagram was created for the office. The major differences of logic is created and realized there.



*Figure 9: Office Communication UML class diagram.*

 Figure 9 is the class diagram for the office. The difference between this system and the pond comes from the client thread. The client thread is a listening channel that sprouts threats as need for each device connected to it, the new threads handles the communication between each device talking to the office system and itself. It uses the same handshake, encryption, ledger code as the pond system.



*Figure 10: Graphical User Activity Diagram.*

By Ahmed Al Qaysi:

The Activity Diagram was the reference used to design the graphical user interface.

## Project Approach

By Ahmed Al Qaysi:

With senior design spanning two semesters, we used the first semester to defining the project requirement then we start to analyze the current system to find the deficiencies. Then we start researching to ensure using the right methods for our design. Each member assigned a part of the project and a due date for the completions. Some of the team member start working on their part as soon as the summer block start others start to implement their part of the project in the beginning of the fall semester.

## Engineering Standards

By Nicholas Holleman:

|  |  |  |  |
| --- | --- | --- | --- |
| **Standard** | **Title** | **Application** | **Relevance** |
| IP65 | **Ingress Protection 65** | Pond electronic enclosure | Safety, Outdoor functionality |
| RSA | Rivest–Shamir–Adleman Algorithm | Method of encrypting and decrypting message from office to pond | Data Integrity |
| P2144.2 | Standard for functional Requirements in Blockchain-based IOT Data Management | Standard used for structural requirements for data ledger | Data Integrity |

*Table 2: Applicable standards.*

## Progress Towards Goals

By Nicholas Holleman:

|  |  |  |
| --- | --- | --- |
| **Feature** | **Objective** | **Status** |
| **GUI** |  |  |
|  | Current water level | Achieved |
|  | Past Water level |  |
|  | Battery life | Achieved |
|  | Predict battery life |  |
|  | Replace LED stripe | To be completed after senior day design |
| **Power Management** |  |  |
|  | Double power life without solar panel | Achieved |
|  | Under voltage and over voltage protection for micro-processor | Achieved |
| **Ledger** |  |  |
|  | Detect altered Hash-code | Achieved |
|  | Detect altered sensor/alert values | Achieved |
|  | Three years of Ledger storage | Achieved |
| **Communication** |  |  |
|  | Transmission efficiency of 50% or greater | Achieved |
|  | Application level encryption | Achieved |
|  | TCP client server application | Achieved |
| **Diagnostic** |  |  |
|  | Implement voltage and current sensor for Solar panel, BMU, Battery | Achieved |
|  | Detect when the antenna powers, and usable | Achieved |
|  | Detect if Office is accepting connections | Achieved |
|  | Detect Voltage if a given sensor reading is to low or High | Achieved |
| Overall | Install system at Ingram | To be Completed after senior day design |

*Table 3: Status of goals.*

The originally proposed deliverables weren’t a though explanation of work that was to be performed by the team. An updated SOW was summited that reflects work stated that would be completed in functional specification, IDR, progress presentation, and FDR.

A small alteration to antenna detection was also adjust. The stated deliverable was too precise on how something was going to be achieved. The overall arching concept is still the same but the method of how its achieved differed. This was done because the method wasn’t possible do. The WIFI antenna simply didn’t have the functionality listed with in the manual.

## Verification

By Nicholas Holleman, 1-8:

By Ahmed Al Qaysi, 9:

Depending on the sub-system each was handled as unit test before assembly to a functional system. Once the system was fully assembled, test was created for the fringes of some systems and for others if they achieve the stated.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tested By:** | | Nicholas Holleman && Andres Oliva | |
| **Test Case Number** | | 1 | |
| **Test Case Name** | | Power management current usage | |
| **Test Case Description** | | The point of this test case is to determine the current draw of the in different operational modes.  Assumptions: µ-controller will be powered at all times | |
| **Item(s) to be tested** | | | |
| 1 | INA219 voltage and current sensors. | | |
| 2 | µ-controller, µ-processor | | |
| 3 | Power management software. | | |
| **Specifications** | | | |
| **Mode of operation** | | | **Output** |
| Sample all sensors, save on µ-processor | | | A log of voltage and milliamps for all modes of operation |
| Power µ-processor | | |
| Power Antenna | | |
| Operate old system | | |
| Sample all sensors with µ-controller | | |
| **Resources Required** | | | |
| 1 | INA219 voltage and current sensors. | | |
| 2 | Lab room space. | | |
| 3 | µ-controller, µ-processor and antenna. | | |
| 4 | Digital multimeter and software assigned to µ-controller and µ-processor to monitor required operational modes. | | |
| **Procedural Steps** | | | |
| 1 | Setup and connect original system to Ina219 and a power source . | | |
| 2 | Run the original software | | |
| 4 | Run logging program on separate µ-processor for 1 hour | | |
| 5 | Connected new hardware to system | | |
| 6 | Run test software that gathers sensor reading, saves and powers down  µ-processor | | |
| 7 | Run test software that gathers sensor reading, saves and transmit on µ-processor | | |
| 8 | Run test software that powers µ-processor and powers down | | |
| 9 | Run Test software that samples sensors on µ-controller | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Tested By:** | | Nicholas Holleman && Andres Oliva | |
| **Test Case Number** | | 2 | |
| **Test Case Name** | | Power management: Time cycle of system | |
| **Test Case Description** | | Test is designed to test time in cycle of each state of system. | |
| **Item(s) to be tested** | | | |
| 1 | INA219 voltage and current sensors. | | |
| 2 | µ-controller, µ-processor | | |
| 3 | Power management software. | | |
| **Specifications** | | | |
| **Input** | | | **Expected**  **Output/Result** |
| Sample all sensors, save on µ-processor | | | A log of time spent in each state |
| Power µ-processor | | |
| Power Antenna | | |
| Transmitting data | | |
| Sample all sensors with µ-controller | | |  |
| **Resources Required** | | | |
| 1 | INA219 voltage and current sensors. | | |
| 2 | Lab room space. | | |
| 3 | µ-controller, µ-processor and antenna. | | |
| 4 | Desk Top power supply | | |
| **Procedural Steps** | | | |
| 1 | Connect system to desk stop power supply | | |
| 2 | Run test software on µ-controller that powers µ-processor and powers it down when boot is finished and monitors the time it takes reports it via serial communication. Sample size 600 cycles, | | |
| 3 | Run test software on µ-controller that cycles sensor readings for 1 hour and reports it via serial communication when finished | | |
| 4 | Run test software on µ-processor that cycles sensor readings, and saves for 1 hour, Record times are saved to a log file | | |
| 5 | Run test software on µ-processor that cycles antenna on to fully functional 100 times, Record times are saved to a log file | | |
| 5 | Run test software on µ-processor that gathers sensor data and transmits in batches of 7 and 1 every 30 seconds, Record times are saved to a log file | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Tested By:** | | Nicholas Holleman | |
| **Test Case Number** | | 3 | |
| **Test Case Name** | | Data ledger storage size | |
| **Test Case Description** | | Test is designed to determine the Storage size requirements of system  Assumptions: New system will only save 12 hours worth do to hours of operation | |
| **Specifications** | | | |
| **Simulate** | | | **Output** |
| Old system | | | Data ledger in block chain format with a certain file size |
| Sample every 30 seconds | | |
| Sample once 1 hour | | |
|  |
| **Resources Required** | | | |
| 1 | INA219 voltage and current sensors. | | |
| 2 | Lab room space. | | |
| 3 | µ-controller, µ-processor and antenna. | | |
| 4 | Desk Top power supply | | |
| **Procedural Steps** | | | |
| 1 | Connect old configuration of system to desk stop power supply | | |
| 2 | Run software and monitor file size before it resets/freeze | | |
| 3 | Run Test software that simulates sensor sample every 30s and 1 hour and inserts them into an empty log and saves. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Tested By:** | | Nicholas Holleman | |
| **Test Case Number** | | 4 | |
| **Test Case Name** | | Ledger corruption detection | |
| **Test Case Description** | | Testing for invalid blocks within the ledger.  Assumptions: Will use Data Set from test case 3 | |
| **Specifications** | | | |
| **Input** | | | **Expected**  **Output/Result** |
| Alter data set current hash-code | | | Expect the system to detect each block with an invalid hash-code or altered sensor value. |
| Alter data set Last hash-code | | |
| Alter data set sensor values | | |
| **Procedural Steps** | | | |
| 1 | Run the server application. | | |
| 2 | Load the ledger. | | |
| 3 | Run detection on each ledger | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Tested By:** | | Nicholas Holleman | |
| **Test Case Number** | | 5 | |
| **Test Case Name** | | Network encryption | |
| **Test Case Description** | | Verifying that the office µ-processor and that the pond µ-processor are communicating properly while encrypting and decrypting packets. | |
| **Item(s) to be tested** | | | |
| 1 | Office µ-processor | | |
| 2 | Antennas | | |
| 3 | Pond µ-processor | | |
| **Specifications** | | | |
| Transmission modes | | | **Expected**  **Output/Result** |
| Invalid key Encrypted Transmission | | Detected at handshake, Refused connection | |
| Invalid Transmissions Hash-code | | Detected bad message Hash-code, Block Ignored | |
| Inserted Invalid Encryption key after handshake | | Failed to Decode message, Message Ignored, disconnected from server | |
| Transmit with Correct key | | Connect accepted, Message Accepted | |
| **Resources Required** | | | |
| 1 | Access to the Ingram Readymix facility. | | |
| **Procedural Steps** | | | |
| 1 | Run applications on office µ-processor and pond µ-processor. | | |
| 2 | Alter encryption keys to fit the need transmission modes | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Tested By:** | | Nicholas Holleman | |
| **Test Case Number** | | 6 | |
| **Test Case Name** | | Transmission Efficiency | |
| **Test Case Description** | | Calculates Efficiency of messages sent | |
| **Item(s) to be tested** | | | |
| 1 | Transmission Efficiency | | |
| **Specifications** | | | |
| **Components** | | | **Expected**  **Result** |
| Hash Last | | | Byte size of message |
| Hash Current | | |
| Time | | |
| Sensor name | | |
| Sensor values | | |
| Alerts names | | |
| Alert values | | |
| Total transmission Size | | |
| **Procedural Steps** | | | |
| 1 | Add up the byte size of each component of the message sent. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tested By:** | | Nicholas Holleman && Andres Oliva | | |
| **Test Case Number** | | 7 | | |
| **Test Case Name** | | Sensor detection | | |
| **Test Case Description** | | Test will verify system sends alerts when sensor readings are either to high to low or frozen. Will sample each sensor configuration 100 times | | |
| **Item(s) to be tested** | | | | |
| 1 | Sensor diagnostic detection | | | |
| **Specifications** | | | | |
| **Configuration** | | | **Expected**  **Result** | |
| Set test voltage above 16 volts | | | | Detected Value was too high for each sensor type and set corresponding alert and reconfigured sensor |
| Set Test voltage below 10 | | | | Detected Value was too low for each sensor type and set corresponding alert, reconfigured sensor |
| Set sample time on sensor below configured requirement (forced frozen sensor reading) | | | | Detected unchanging sensor value and attempted to reconfigure sensor, then power down and up sensor |
| **Procedural Steps** | | | | |
| 1 | Run application on both pond µ-processor and office µ-processor | | | |
| 2 | Check alerts that appear in log, and on office system | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tested By:** | | Nicholas Holleman | | |
| **Test Case Number** | | 8 | | |
| **Test Case Name** | | Antenna detection | | |
| **Test Case Description** | | Test will Verify It detects the WIFI antenna, and that the correction mechanism work. | | |
| **Specifications** | | | | |
| **Configuration** | | | **Expected**  **Result** | |
| Unplugged | | | | Powers antenna |
| Non-responsive | | | | Cycles whole system |
| Proper operation | | | | Does nothing |
| **Procedural Steps** | | | | |
| 1 | Run application on both pond µ-processor and office µ-processor | | | |
| 2 | Check alerts that appear in logs of pond system, record observed reactions | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Tested By:** | | Ahmed Al Qaysi | |
| **Test Case Number** | | 9 | |
| **Test Case Name** | | GUI | |
| **Test Case Description** | | This test will conform the GUI receiving the data and display the correct output. | |
| **Item(s) to be tested** | | | |
| 1 | Size, position, width, and height of the elements. | | |
| 2 | Testing of the error messages that are getting displayed. | | |
| 3 | Execute the intended functionality. | | |
| 4 | Two clicks max to go anywhere. | | |
| **Specifications** | | | |
| **Input** | | | **Expected**  **Output/Result** |
| Data contain readings for battery life, sensors, and error reading. | | | Should display the correct battery life, sensor reading and a warning message. |
| **Resources Required** | | | |
| 1 | Two Raspberry Pi’s | | |
| 2 | Touch screen display | | |
| 3 | Python | | |
| **Procedural Steps** | | | |
| 1 | Run the application on one of the Raspberry Pi #1. | | |
| 2 | Click on all the tap to check functionality. | | |
| 3 | Sending data from Raspberry Pi #2. | | |
| 4 | Check if the GUI read the data. | | |
| 5 | Check if the GUI have the right output associate with data. | | |

*Table 4: Test cases.*

## Characterization Results

By Nicholas Holleman, 1-8:

By Ahmed Al Qaysi, 9:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case #1 & #2** | | | | | | | | | | | | | | |
| **Requirements** | | | | | | | | | | | | | | |
| **Twice the hours of operation as the System currently in place** | | | | | | | | | | | | | | |
| **Sample System every 30 seconds** | | | | | | | | | | | | | | |
| **Results of Test Case #1** | | | | | | | | | | | | | | |
| Input/Devices on | | Results | | | | | | | | | | | | |
| Voltage | | | Current(mA) | | | STD(mA) | | | | | | Samples |  |
| µ-processor, µ-controller, Sensors | | 11.91 | | | 430 | | | 21.23 | | | | | | 1200 |  |  | | |  |  |
| µ-processor, µ-controller, Sensors, Transmitting | | 11.91 | | | 445 | | | 46.23 | | | | | | 1200 |  |
| µ-controller, Sensors | | 12.35 | | | 6.23 | | | 1.2 | | | | | | 4231 |  |
| µ-processor | | 12.12 | | | 324 | | | 73.6 | | | | | | 600 |  |
| Old System | | 12.17 | | | 459 | | | 35.78 | | | | | | 1200 |  |
| **Results of Test Case #2** | | | | | | | | | | | | | | |
| Operational modes | | Results | | | | | | | | | | | | |
| Means(seconds) | | | | | STD (seconds) | | | | | | Samples | |  |
| Powered µ-processor and sample sensors | | 1.34 | | | | | 0.41 | | | | | | 500 | |  |  |
| Transmitting data | | 8.89 | | | | | 3.26 | | | | | | 100 | |  |  |
| Sensor cycle On µ-controller | | 28 | | | | | .485 | | | | | | 500 | |  |  |
| Powering µ-processor | | 16.23 | | | | | 2.48 | | | | | | 100 | |  |  |
| Powering antenna | | 108.48 | | | | | 1.236 | | | | | | 100 | |  |  |
| **Combined Analytic of Test Case #1 & #2** | | | | | | | | | | | | | | |
| Operational modes | | Results | | | | | | | | | | | | |
| Watts/Hour | | | Watts/Day | | | Days | | | | | | |
| Old System | | 5.58 | | | 134.06 | | | 1.79 | | | | | | |
| Transmitting ever 30s during hours of operation | | 5.19 | | | 62.37 | | | 3.84 | | | | | | |
| Transmitting once a day and sampling every once an hour  During operational hours | | 0.25 | | | 3.03 | | | 79.09 | | | | | | |
| **Compliance** | | | | | | | | | | | | | | |
| **minimum requirement** | | **Result** | | | | | | | **Achieved** | | | | | |
| 30 Second sensor sample | | 28 seconds | | | | | | | Exceed Specification | | | | | |
| 1.79 Days | | 3.84-79 days | | | | | | | Exceed Specification | | | | | |
|
| **Test Case #3 & #4** | | | | | | | | | | | | | | |
| **Requirements** | | | | | | | | | | | | | | |
| **Three years of data ledger storage** | | | | | | | | | | | | | | |
| **Detect alterations within Hash-code and sensor values** | | | | | | | | | | | | | | |
| **Results of Test Case #3** | | | | | | | | | | | | | | |
| Storage Capacity  Operational modes | | Results | | | | | | | | | | | | |
| Sample rate | | | | Duration of log | | | | | | Size (Kbyte) | | |
| Old System | | 30s | | | | 10 mins | | | | | | 1 | | |
| New System | | 3600s | | | | 12 hours/1 day | | | | | | 12 | | |
| New System | | 30’s | | | | 12 hours/1 day | | | | | | 1,491 | | |
| **Results of Test Case #4** | | | | | | | | | | | | | | |
| Hash-code | Detected | | | | | | | | | | | | | |
| Sensor Values | Detected | | | | | | | | | | | | | |
| **Combined Analytic of Test Case #3 and #4** | | | | | | | | | | | | | | |
| Storage Capacity | | Results | | | | | | | | | | | | |
| Months (Kbyte) | | | | Years (MByte) | | | | | | 3 Years | | |
| 3600s samples with no rain | | 360 | | | | 4.32 | | | | | | 12.96 | | |
| 30s samples permanently during hours of operation | | 44,730 | | | | 536.76 | | | | | | 1610.28 | | |
| **Compliance** | | | | | | | | | | | | | | |
| **Minimum requirement** | | **Result** | | | | | | | | | | **Achieved** | | |
| Three years of data ledger storage | | 1.610 GByte used of 16 GByte | | | | | | | | | | Meet  Specification | | |
| Detect alterations within Hash-code and sensor values | | Passed | | | | | | | | | | Exceed Specification | | |
|
| **Test Case #5** | | | | | | | | | | | | | | |
| **Requirements** | | | | | | | | | | | | | | |
| Ignore Invalid Transmissions | | | | | | | | | | | | | | |
| **Results of Test Case #5** | | | | | | | | | | | | | | |
| Transmission modes | | Results | | | | | | | | | | | | |
| Invalid key Encrypted Transmission | | Detected at handshake, Refused connection | | | | | | | | | | | | |
| Invalid Transmissions Hash-code | | Detected bad message Hash-code, Block Ignored | | | | | | | | | | | | |
| Inserted Invalid Encryption key after handshake | | Failed to Decode message, Message Ignored, disconnected from server | | | | | | | | | | | | |
| Transmit with Correct key | | Connect accepted, Message Accepted | | | | | | | | | | | | |
| **Compliance** | | | | | | | | | | | | | | |
| **minimum requirement** | | | **Result** | | | | | | | **Achieved** | | | | |
| Ignore Invalid Transmissions | | | Only accepts Encrypted message with corresponding hash-code | | | | | | | Meet  Specification | | | | |
| **Test Case #6** | | | | | | | | | | | | | | |
| **Requirements** | | | | | | | | | | | | | | |
| Transmit efficiency of 50% or greater. | | | | | | | | | | | | | | |
| **Results of Test Case #6** | | | | | | | | | | | | | | |
| Parts of transmitted parts | | Results | | | | | | | | | | | | |
| Bytes | | | | | | | | | | | | |
| Hash Last | | 8 | | | | | | | | | | | | |
| Hash Current | | 8 | | | | | | | | | | | | |
| Time | | 17 | | | | | | | | | | | | |
| Sensor name | | 30 | | | | | | | | | | | | |
| Sensor values | | 35 | | | | | | | | | | | | |
| Alerts names | | 35 | | | | | | | | | | | | |
| Alerts values | | 40 | | | | | | | | | | | | |
| Total transmission Size | | 256 | | | | | | | | | | | | |
| **Analytics of Test Case #6** | | | | | | | | | | | | | | |
| Results | | | | | | | | | | | | | | |
|  | | Useful | | | | Over head | | | | | | Efficiency | | |
| Transmission | | 165 | | | | 91 | | | | | | 64 | | |
| **Compliance** | | | | | | | | | | | | | | |
| **minimum requirement** | | **Result** | | | | | | | | | | **Achieved** | | |
| Transmit efficiency of 50% or greater. | | 64% Efficient | | | | | | | | | | Exceeds  Specification | | |
| **Test Case #7** | | | | | | | | | | | | | | | | | |
| **Requirements** | | | | | | | | | | | | | | | | | |
| Detect Abnormal High Sensor Value | | | | | | | | | | | | | | | | | |
| Detect Abnormal Low Sensor Value | | | | | | | | | | | | | | | | | |
| Detect Unchanged Sensor Value | | | | | | | | | | | | | | | | | |
| **Results of Test Case #7** | | | | | | | | | | | | | | | | | |
| Sensor | | Results | | | | | | | | | | | | | | | |
| Value to High | | Detected Value was to high for each sensor type and set corresponding alert and reconfigured sensor | | | | | | | | | | | | | | | |
| Value to Low | | Detected Value was to low for each sensor type and set corresponding alert, reconfigured sensor | | | | | | | | | | | | | | | |
| Value to Frozen | | Detected unchanging sensor value and attempted to reconfigure sensor, then power down and up sensor | | | | | | | | | | | | | | | |
| **Compliance** | | | | | | | | | | | | | | | | | |
| **minimum requirement** | | | **Result** | | | | | | | **Achieved** | | | | | | | |
| Detect Abnormal High Sensor Value | | | **Passed** | | | | | | | Meet  Specification | | | | | | | |
| Detect Abnormal Low Sensor Value | | | **Passed** | | | | | | | Meet  Specification | | | | | | | |
| **Test Case #8** | | | | | | | | | | | | | | | | | |
| **Requirements** | | | | | | | | | | | | | | | | | |
| Detects Remote pond antenna is operational status | | | | | | | | | | | | | | | | | |
| **Results of Test Case #8** | | | | | | | | | | | | | | | | | |
| **Antenna State** | | **Results** | | | | | | | | | | | | | | | |
| Unplugged | | Powers antenna | | | | | | | | | | | | | | | |
| Non-responsive | | Cycles whole system | | | | | | | | | | | | | | | |
| Proper operation | | Does nothing | | | | | | | | | | | | | | | |
| **Compliance** | | | | | | | | | | | | | | | | | |
| **minimum requirement** | | | **Result** | | | | | | | **Achieved** | | | | | | | |
| Detects Remote pond antenna is operational status | | | **Powers, Cycles system, Does nothing** | | | | | | | Meet  Specification | | | | | | | |
| Detect Unchanged Sensor Value | | | **Passed** | | | | | | | Meet  Specification | | | | | | | |
| **Test Case #9** | | | | | | | | | | | | | | | | | |
| **Requirements** | | | | | | | | | | | | | | | | | |
| Receiving the data and display the correct output. | | | | | | | | | | | | | | | | | |
| **Results of Test Case #9** | | | | | | | | | | | | | | | | | |
| GUI | | | | Results | | | | | | | | | | | | | |
| Battery level | | | | Icon changed | | | | | | | | | | | | | |
| Water level | | | | Water level graph changed | | | | | | | | | | | | | |
| Alerts | | | | Message received and displayed | | | | | | | | | | | | | |
| **Compliance** | | | | | | | | | | | | | | | | | |
| **minimum requirement** | | | **Result** | | | | | | | | **Achieved** | | | | | | |
| Data contain readings for battery life, sensors, and error reading. | | | display the correct battery life, sensor reading and a warning message | | | | | | | | Meet  Specification | | | | | | |

*Table 5: Results of test cases.*

## Deficiencies

By Ahmed Al Qaysi:

Our team was unable to let the user access the remote retention pond through a phone application or a web browser which was one of our stretch coals for our project. We will need approximately about 6 weeks to design, create and test the web browser.

## Iterations and Redefinitions

By Nicholas Holleman:

Two redefinitions accrued during the course of the project. A major change was in software used to create the graphical user interface. At first the project was to use KIVY. The team member that first attempted to use the KIVY found the lack of documentation and examples too difficult to overcome. The choice four weeks into the project was to change to PyQt.. This change facilitated an increased in productivity on this area of the project.

The second redefinition of the project came from being to specific on how something was to be accomplished. The antenna manual suggested that it could connected to via a terminal to view and adjust parameters on the antenna. The end result was the antenna could be connected to but no commands for the system are listed that allowed for changing or viewing of antenna functions. So to determine if the antenna was functional a request to the web socket on the antenna is used to determine if the antenna is powered and operational.

Both instance of re-definition re-confirmed the need to be flexible in design a product. Meaning don’t get be overly obsessed doing something in a certain fashion.

# Constraints

## Budgetary

By Nicholas Holleman:

For our project, we were allotted a $500 budget for supplies in order to complete this project. The budget was the major concern of the project. The small budget and goals that needed to be accomplished didn’t initially align. After the team took to the aspect of re-tooling the system to fit the design needs, meaning the team would simply replace the software and consider each hardware addition to what it adds in terms of functionality to cost, the end goal became manageable.

Based on a re-tooling frame of mind, we decided to use a microcontroller and voltage current sensors to satisfy our Power Management requirement and 7” LCD screen to obtained data visualization. The budget for these items fell within the allocated budget.

## Design Feasibility

By Ahmed Al Qaysi:

The route our team took with using a Raspberry Pi 3B was for compatibility purposes from

past teams. Having said this, this project is very flexible in the sense that it can be integrated with many types of sensors, microcontrollers and microprocessors. For example, using the Sleepy Pi Arduino to handle the Raspberry Pi sleeping mode. Consequently, for this project to move forward, our team took the approach in augmenting to previous work done by previous teams in order to have same hardware and software language.

## Manufacturability

By Andres Oliva:

When putting together this project our team did not go through any severe manufacturability constraints. This was possible with the help from the previous team. Majority of the parts used to create this product were purchased to protect all components that would be implemented. This project required substantial programming skills to complete, nevertheless the rest are parts that can be purchased such as the micro-controller, microprocessors and the sensors. All these electrical components can be placed inside a waterproof casing to protect from the rain.

## Maintainability

By Ahmed Al Qaysi:

The system that will be implemented will require minimal maintenance. This will consist of cleaning the solar panel once a month for optimal charging performance, and battery replacement if necessary or when alerted otherwise. No personal support will be provided for this project, but a Github repository will be publicly available with all software and build of materials.

## Environmental

By Ahmed Al Qaysi:

Since we augmented this project, we had to make sure that anything we used have to be able to the environmental requirement. The server Pi, the base station office unit, will experience normal room temperature and face no environmental challenges. The client Pi, the outdoor unit, will be designed to operate within a temperature range of 0 to 120 degrees and under adverse weather conditions.

## Health and Safety

By Andres Oliva:

During the implementation process of the project, there were a few health and safety constraints that had to be taken into consideration. The first safety concern that had to be considered was upon entry to the Ingram Readymix concrete facility. Since the company handles and operates heavy machinery, all members of the team were instructed to wear proper protective equipment to enter through the facility to the outdoor water retention pond. Continuing with safety concerns, the team had to ensure that there were no lose wires since the system being handled is electrical. Everything that was implemented and upgraded had to fit into a water-proof case installed by the previous team at the water retention pond. Lastly, the final safety concern that was considered throughout the design of this project was to ensure that the data was not altered or tampered with. Nicholas Holleman designed a system that can keep data safe through the implementation of encryption.

## Social

By Nicholas Holleman:

The system was designed had the constraint of who the audience of the product was for. This means the interaction with the system had to be boiled down to is the simple enough for someone to understand. How many clicks in the GUI is too many clicks to access a certain portion of the graphical user interface?

This meant any type of diagnostic within the system need to system handled. If the system couldn’t resolve the correct the corresponding error created, it need to alert the user to what exactly was wrong and what need to be done to address the issue. The solar panel not producing a high enough voltage should promote the user to clean the solar panel. Or if the system couldn’t find antenna it notifies the user to check the LEDs on the antenna to determine what’s truly wrong with it.

# Budgets

By Ahmed Al Qaysi:

|  |  |  |
| --- | --- | --- |
| Item | proposed | Actual |
| Sleepy pi | $43.57 | $49.95 |
| 3X INA219 Sensor | $10.99 | $9.99 |
| Solder kit | $40 | $0 |
| IC's | $200 | $0 |
| PC board | $75 | $0 |
| Case for The Raspberry Pi LCD screen |  | $19.99 |
| HUAWEI LTE USB Stick |  | $79.99 |
| Soracom Air Global IoT SIM Card |  | $5 |
| Total | $369.56 | $164.92 |

*Table 6: Budget proposed versus actual spent.*

Our actual budget is different than the proposed with the amount of $205.20 less because we found that we don’t need to purchase solder kit and IC`s. We purchased the Case for the Raspberry Pi LCD screen. After the summer break when we went back, we found that all the parts that we received from the previous team was disappeared. And buying the HUAWEI LTE USB Stick and Soracom Air Global IoT SIM Card these two needed to work on one of our stretch coals.

# Work Schedule

By Ahmed Al Qaysi:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task** | **Proposal due** | **Actual due** | **IDR** | **Compilation** |
| **Start** | **1/28/2019** | **1/28/2019** | **All** | **Yes** |
| Team formed | 1/28/2019 | 1/28/2019 | All | Yes |
| **SOW** | **2/19/2019** | **2/19/2019** | **All** | **Yes** |
| Define project requirement | 2/05/2019 | 2/05/2019 | All | Yes |
| Research Ideas | 02/13/2019 | 02/13/2019 | All | Yes |
| Write SOW section | 02/18/2019 | 02/18/2019 | All | Yes |
| Turn SOW | 02/19/2019 | 02/19/2019 | Ahmed | Yes |
| **Design** | **05/07/2019** | **05/07/2019** |  | **Yes** |
| Power Management | 02/26/2019 | 02/26/2019 | Andres | Yes |
| GUI | 02/26/2019 | 02/26/2019 | Ahmed | Yes |
| Log Ledger | 02/26/2019 | 02/26/2019 | Nicholas | Yes |
| Self-Diagnose | 02/26/2019 | 02/26/2019 | Nicholas | Yes |
| Functional specification | 03/13/2019 | 03/13/2019 | All | Yes |
| Initial Design Review | 04/01/2019 | 04/01/2019 | All | Yes |
| Update Functional Spec | 04/17/2019 | 04/17/2019 | All | Yes |
| Labor-Cost-Schedule | 04/22/2019 | 04/22/2019 | All | Yes |
| Poster | 04/24/2019 | 04/24/2019 | All | Yes |
| Test plan | 05/01/2019 | 05/01/2019 | All | Yes |
| Senior design day | 05/07/2019 | 05/07/2019 | All | Yes |
| Summer break | 08/28/2019 | 08/28/2019 | All | Yes |
| **Assembling & Software Development** | **10/18/2019** |  | **All** | **Yes** |
| **Blockchain** | **10/17/2019** | 11/17/2019 | Nicholas | Yes |
| Block | 09/16/2019 | 09/16/2019 | Nicholas | Yes |
| Chain | 10/08/2019 | 10/23/2019 | Nicholas | Yes |
| Test | 10/17/2019 | 11/17/2019 | Nicholas | Yes |
| **GUI** | **10/18/2019** | **11/19/2019** | **Ahmed** | **Yes** |
| Temple | 09/06/2019 | 10/03/2019 | Ahmed | Yes |
| Battery screen | 09/24/2019 | 10/23/2019 | Ahmed | Yes |
| Diagnostic screen | 10/11/2019 | 11/07/2019 | Ahmed | Yes |
| Test | 10/18/2019 | 11/19/2019 | Ahmed | Yes |
| **Sensors** | **10/18/2019** | **11/25/2019** | Andres | **Yes** |
| Physical implement sensor | 09/04/2019 | 10/20/2019 | Andres | Yes |
| Sensor analysis | 10/01/2019 | 11/01/2019 | Andres | Yes |
| Test | 10/18/2019 | 11/25/2019 | Andres | Yes |
| **Diagnostic** | **10/18/2019** |  | Nicholas | Yes |
| Sensor | 09/09/2019 | 10/27/2019 | Nicholas | Yes |
| Network/OS | 09/25/2019 | 11/01/2019 | Nicholas | Yes |
| Blockchain | 10/10/2019 | 10/23/2019 | Nicholas | Yes |
| Test | 10/18/2019 | 10/23/2019 | Nicholas | Yes |
| **Progress** | **10/17/2019** |  | **All** | **Yes** |
| Progress presentation | 10/17/2019 | 09/23/2019 | All | Yes |
| Demo day | 10/17/2019 | 10/07/2019 | All | Yes |
| **Testing** | **12/06/2019** |  | Nicholas | **Yes** |
| System test | 11/18/2019 | 12/06/2019 | Nicholas | Yes |
| FDR | 12/04/2019 | 11/11/2019 | All | Yes |
| Update poster | 12/03/2019 | 12/01/2019 | All | Yes |
| Senior design 2 Day | 12/06/2019 | 12/06/2019 | All | Yes |

*Table 7: Work schedule proposed versus actual schedule.*

# Personnel Interactions

## Teamwork

By Andres Oliva:

Project responsibilities were split between all team members. The designated responsible individuals were chosen for specific portions of the project based on their areas of experience. Starting with the project manager, Ahmed Al Qaysi was chosen to design and implement the graphical user interface (GUI) for the visualization portion of this project. With his experience in computer engineering, this was something agreed amongst the group for the development and implementation. Next, the engineer Andres Oliva was assigned with the task of implementing a power management system with his experience in micro-and nano technology and the use of technical skills to upgrade new components to the existing system. Lastly, the engineer Nicholas Holleman was assigned the task of creating a self-diagnostic system, the implementation of data ledger, and implementation of securing the data transmissions.

## Mentorship

By Nicholas Holleman:

Sponsor being a concreate plant didn’t have much technical advice. This isn’t to say they didn’t help; they just provide us with the opportunity to fulfill their request for improvements to the system how we so saw fit. This caused the team to stop and continual ask “How does this improve the system and is it worth the cost?”.

The faculty advisor is a different issue all together. During design I the team meet with McClellan 6 times, each meeting helped define the project to what it was when the initial design review accrued. McClellan gave good feedback to how we proposed to alter or adjust the system. Design II was a little different the team held 2 meetings. Each meeting gave a clarity in different manners. One meeting gave clarity on how to handle certain questions that could address from a design choice and how to relate them back the function specification, and SOW. The other gave feedback what he thought of the overall system and how it functioned. For the technical aspect of the project because of the limited overall interactions no technical questions from McClellan were asked of him or from him thus none were given.

# Ethics

By Andres Oliva:

Throughout the course of this year long project, the team had to take ethical theories into consideration. Some of these theories that applied to our project appear under the IEEE code of ethics. These reference honesty and realism in stating claims based on available data; to hold paramount the safety, health and welfare of the public, and to disclose promptly factors that may danger the public or the environment. Our team followed these ethical theories especially when attempting to help others understand the implications of conventional and emerging technology that would be applied to the facility.

# Summary & Conclusions

By Andres Oliva:

The Ingram Readymix Secured Network is a project that focuses on creating a power management system, a blockchain data ledger and a graphical user interface (GUI). These features were designed and created by a team of three engineers.

The project was sponsored by the Ingram Readymix cement company and can create a sustainable system using a Sleepy Pi micro-controller that is attached to the Raspberry Pi microprocessor. The INA219 voltage and current sensors are also attached to the micro-controller to calculate and read values that are used for analysis and data visualization. The project is intended to allow remote access and security for the overall system located at the facility. The group provides security through the implementation of blockchain and the application of encryption. The team was successful in reaching desired goals for the project. The power management subsystem showed success and was implemented to extend the battery life. The results assured that the micro-controller was working the way that our team intended it to. Secured communication between the office and the retention pond was enhanced with a layer of encryption applied which also showed success when implemented. The GUI was essential when our team wanted to test various functions that would be implemented at the facility. Some of these functions consisted of the battery status, and the water levels to protect the Ingram Readymix company from possible fines.

# Discussion

## Academic Preparation

By Ahmed Al Qaysi:

|  |  |  |
| --- | --- | --- |
| **Course No.** | **Core knowledge** | **Specific knowledge incorporated by team** |
| EE 3350 (Electronics I) | Design and analysis of active devices and equivalent circuits | Not designing circuit. Using pre-made devices. |
| EE 3370 (Signals and Systems) | Frequency domain representation of signals and frequency response, transfer functions | Nyquist theorem for sampling sensor data. |
| EE 3420 (Microprocessors) | Principles of operation and applications of microprocessors | I2C communication. |
| EE 4352 (Introduction to VLSI Design) | Analysis and design of CMOS integrated circuits | Not designing or analyzing integrated circuits |
| EE 4370 (Communications Systems) | Transmission of signals through linear systems, analog and digital modulation, and noise | Frequency domain analysis of sensor data, Signal to Noise ratio |

*Table 8: Academic preparation table.*

## Lessons Learned

By Andres Oliva:

Throughout the duration of this project, there were many lessons to be learned. This consisted of teamwork, communication, work ethic, time management, and the list can go on. Everyday something new was learned, whether it be regarding the project, or regarding the members of the team. Of course, there were many struggles, but this is what made the team stand strong and achieve common goals.

## Soft Skills

By Nicholas Holleman:

The Top soft skill the class provide for the team in the form of presentation skills, and communication. Presentation skills took the team a while to understand. The activity of giving a presentation was the best teacher. The feed-back from professors was helpful when it was relevant to content and explained in meaningful way. Feed-back from students was that different, students seemed to praise presentations and didn’t understand what they needed to criticize to give meaningful feed-back.

Communication as a team was a challenge. The team had to rely on text message this term due to scheduling conflicts. Meaning each message had to be consider from a standpoint of being profession and viewed as sarcasm/humor wouldn’t have been perceived by the other party. This skill was further developed while writing the Final report. Comments about what someone was trying to say or asking if people had advise need for the section was beyond helpful.

## Schedule Deviations

By Ahmed Al Qaysi:

There were two main problems that cause the project to face a delay, first was creating a functional GUI and this happened due to the unexperienced of creating GUI and this lead to make a wrong design choice, hade change the library that use to build the GUI. Second was the power management and this due to the complicity. We fix both issues and come back to be on track by putting more hours and all the team worked on solve these issues.

## Staffing

By Andres Oliva:

This project was staffed adequately with three members assigned. The group consisted of a project manager with a specialty in computer engineering; two engineers, both with specialties in micro and nano technology. The Ingram Readymix: Secured Network project is focused around these types of disciplines which required substantial programming and technical skills to install additional components such as micro-controllers and sensors. The shared knowledge allowed the team to achieve desired goals with the aspiration to satisfy all requirements.

## Final Observations

By Nicholas Holleman:

If the team was to do the project over again it would require a slightly larger budget and not come from the prospective of a re-tooling or augmenting the previses system. Meaning hardware would have been slightly different. First the BMU would have been one that provides the functionality of communication to and outside network or over SPI/I2C. This would have allowed time to allocated toward something that expands the functionality of the system.

The project would have also spent capital on making the device a Internet of things (IOT) device instead of a local system. So instead of having WIFI antenna the system would have a 3G/4G modem. This would allow device information to be shared anywhere in the world, like off site at central Ingram concreate office that could monitor multi-system at once. It would also allow each system to send text messages to personal if it couldn’t contact the office when water exits via the storm drain.

With a micro-controller and micro-processor currently present and complicated design a single micro-processor would have been used. A few micro-processors exist that could fulfill the needs but didn’t have the end user-support are team required to handle encryption. With time having been freed from creating a monitoring system for the battery system the time would be spent here porting the encryption library required. This would have reduced the complexity and long-term cost associated with manufacturing of the system.

Cost being a factor in all projects, reducing the size of the solar panel could now be achieved. Due to the addition of a power management system to the device, the team would then decide the required size of the solar panel based on that data.

# Acknowledgments

By Nicholas Holleman:

The Ingram Readymix: Secured network team would like to thank are sponsor Ingram ReadyMix for having the opportunity to design a system for them. The Team would also like to Thank Dr. McClellan for all the frank conversation, that lead to simple advise we can use in the future for are careers. Finally, the team would like to thank Mr. Welker for putting put with all the small meaningless questions and well-defined feed-back given on any assignment turned in.

# References

[1] Tycon Systems EZ-Bridge® LT+ Outdoor Wireless Ethernet Bridge.

*Data Sheet.* [*https://tyconsystems.com/documentation/Spec%20Sheets/EZ-Bridge-LT+%20DataSheet.pdf*](https://tyconsystems.com/documentation/Spec%20Sheets/EZ-Bridge-LT+%20DataSheet.pdf)

*[2]* Raspberry Pi 3 Model B+

[*https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf*](https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf)

[3] GENERAL PERMIT TO DISCHARGE UNDER THE TEXAS POLLUTANT DISCHARGE ELIMINATION SYSTEM. <https://www.tceq.texas.gov/assets/public/permitting/stormwater/txr050000.pdf>

[4] Raspbian: Raspberry Pi operating system. <https://www.raspberrypi.org/documentation/raspbian/>

[5] Python 3.X (programming language). <https://docs.python.org/3/>

[6] PyQt . <https://wiki.python.org/moin/PyQt/>