# Low-Cost Wireless High Water Detection System

A Design Project Report

Presented to the School of Electrical and Computer Engineering of Cornell University

in Partial Fulfilment of the Requirements for the Degree of

Master of Engineering, Electrical and Computer Engineering

Submitted by

Chen-hang Yen

MEng Field Advisor: Bruce Land

Date: January, 2014

# Abstract

Master of Engineering Program
Cornell University
Design Project Report

**Project Title:** Low-Cost Wireless High Water Detection System

**Author:** Chen-hang Yen

**Abstract:**

This design project aims to create an inexpensive flood detection system to monitor rising water in remote locations or residential areas. The high water detection system divides into two parts: water sensing unit and data display unit. Both subsystems are based on the ATmega 328P microcontroller and they are communicating wirelessly via radio frequency (RF) transceivers. Additionally, several custom-built modules, including water sensors, charging regulator, and status board, are designed to support moisture detection, power management, and information display.

# Executive Summary

This design project is created to reduce flood damages and to combat increasing flood risks in remote areas, particularly places without cellphone signal coverage. The principle of most high water detection systems is similar, which includes moisture sensing and communication. The goal of the project is to devise a wireless flood detection system with low power consumption and simple design.

The water detection system contains a transmitter subsystem and a receiver subsystem. The transmitter subsystem is consisted of the following modules:

- Low-power microcontroller board – 3.3 volt Arduino Pro Mini board supervises all transmitter system activities
- RF transmission element – 2.4 GHz transceiver relays raw data to the receiver subsystem
- Power management board – Customized MCP73877-based board to regulate power supply and charging process
- Water sensors – Basic water sensing components to detect rising water at multiple levels

The self-powered transmitter subsystem can be installed at most outdoor locations.

On the other hand, the receiver subsystem requires external power source. It is comprised of the following modules:

- Stable microcontroller board – 5 volt Arduino Uno board processes incoming data and triggers appropriate responses
- RF transmission element – 2.4 GHz transceiver captures raw data and sends to microcontroller for processing
- Information display board – LED-based notification board provides latest water level update to users
- Siren component – Plain buzzer offers acoustic warning to signal any flood risks

Overall, the cost of materials is less than $60 and it delivers comparable functionalities as many expensive flood detection devices.

# Table of Contents

# List of Figures

# 1 Introduction

Due to rapid climate change in recent decades, an increase in the severity of flood-related damages is observed. This causes serious destruction to residential properties and it also threatens public safety, particularly residents in the coastal regions or in the areas with heavy rainfalls. The Federal Insurance and Mitigation Administration (FEMA)'s National Flood Insurance Program (NFIP) estimates that total losses due to six-inch flood are approximately $20,000 per 1,000 square foot home [1]. Although several commercial flood warning systems are currently available, many of them are either expensive or unable to identify multiple water levels. In fact, some water detection devices are triggered by a single event and their alerts are broadcasted via a buzzer. However it is often too late for people to protect their belongings and evacuate to safe ground if their flood warning appliance is solely activated by a certain water level without pre-flood warning.

The goal of this design project is to create a low-cost wireless high water detection system that senses rising water in real time and determines any potential flash floods. The current design includes a solar-powered water recognition system wirelessly transmitting sensor data to a receiver system via radio frequency (RF) transceivers.

## 1.1 Design Alternatives

There are numerous ways to detect water levels and notify flood risk. Callaway, Frechette, and Trapp's winning flood warning station used two sensors to detect specific water levels and triggered a strobe siren when flash flood was identified [2]. National Weather Service (NWS)'s flood warning systems manual offered a range of high water notification solutions which can be implemented at various locations, including reservoirs, streams, and local communities [3]. A wide selection of commercial water detection products were also obtainable online [4]-[6].

Despite countless design possibilities, flood recognition systems typically consist two components – identification and communication [3]. This design project aims to provide an inexpensive method to advising possible floods for residents in the remote areas where cellphone signal coverage is usually absent.

# 2 System Design

The solar-powered flood detection system with RF transceiver is divided into two parts, transmitter system and receiver system. They are crafted to address the following design considerations:

- Total cost of materials is less than $70
- Transmitter system needs to be self-powered with low power consumption
- Water sensors need to detect multiple water levels
- Sensor data can be transmitted and displayed via wireless medium
- A display module is required to show water levels and transmitter status in real time
- A high water warning indicator to inform user any flood risk

## 2.1 High-Level Design

Figure 1 demonstrates the high-level system design concept. A portable transmitter system with solar charging feature wirelessly sends water level data to a stationary receiver system. As water rises, the mounted sensors detect appropriate water level, then the transmitter system complies and relays raw data via a RF module, and finally the receiver system interprets incoming data signal and displays real-time information on a dashboard.

In this project, the transmitter system is powered by a solar panel coupled with a high-capacity Lithium polymer (LiPo) battery to reduce touch maintenance. A compact RF package is also employed to enhance wireless transmission distance.

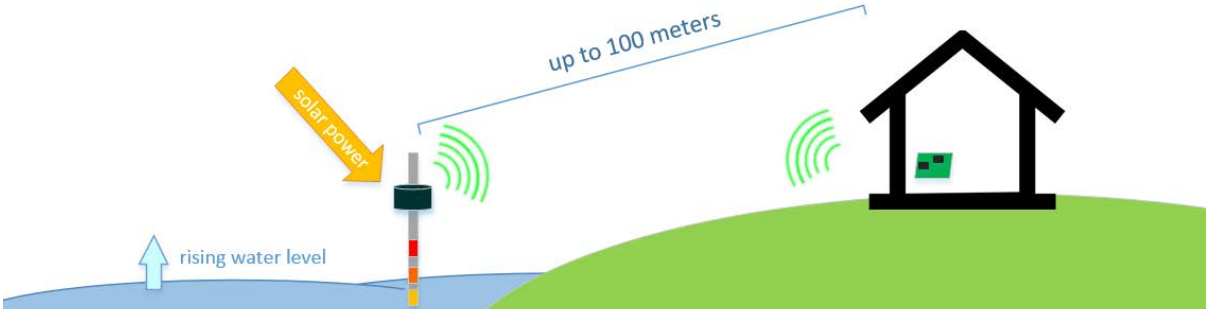


**Figure 1:** Design overview

To reduce maintenance requirement, the transmitter system utilizes a low-power microcontroller board, simple water detection circuits, sturdy RF transceiver package, and a power management module to regulate distribution of harnessed solar power. Figure 2 shows the transmitter system flowchart. Simplicity and power saving are two key focuses in designing the transmitter system. Therefore, microcontroller is operating at the minimum clock speed and is directly transmitting raw data, gathered from water sensor module, to the receiver system.

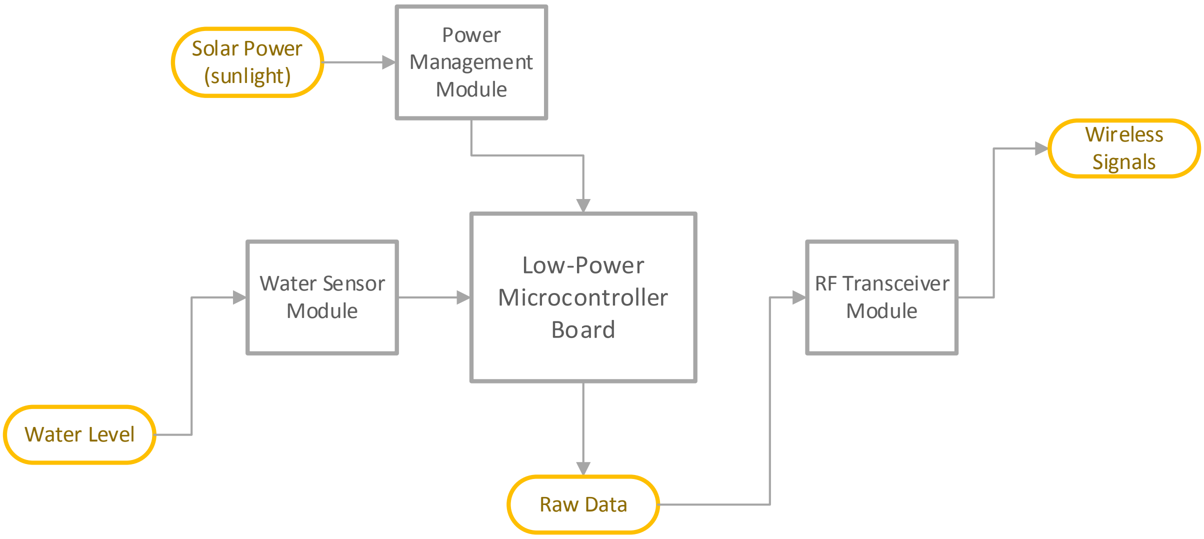


**Figure 2:** Transmitter system flowchart

Reliable data interpretation and real-time event notification are crucial to the receiver system design. Figure 3 shows the receiver system flowchart. In this part, the receiver microcontroller board processes all incoming data and illuminates corresponding light emitting diodes (LEDs) on the information display board to report current water level and transmitter status. The flood warning module is activated when the system detects persistent high water level. Also, the receiver system automatically resets the display module as soon as water level recedes below the sensor line.

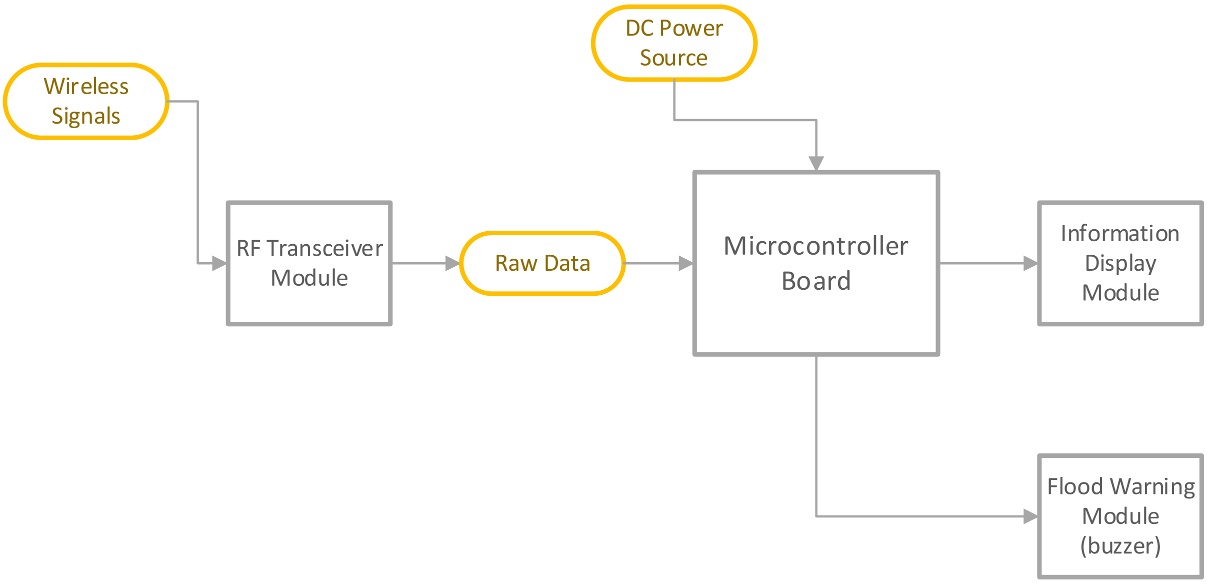


**Figure 3:** Receiver system flowchart

## 2.2 Hardware Design

The following subsections describe hardware design and materials selection.

### *2.2.1 Low-Power Microcontroller Boards*

Microcontroller is the brain of this design project. Microcontroller, a miniature computing chip, enables electronic systems to perform complex logical calculations. Among various microcontroller integrated circuits (ICs), Arduino microcontroller is chosen to be the main processor of both transmitter and receiver components.

Equipped with the ATmega 328P, Arduino microcontroller boards are known for their versatility, wide range of applications, and well-documented software libraries. Many believe there is an Arduino board for any circuit project.

All Arduino microcontroller boards are operating at either 5 volts or 3.3 volts. Figure 4 illustrates basic anatomy of an Arduino microcontroller board.

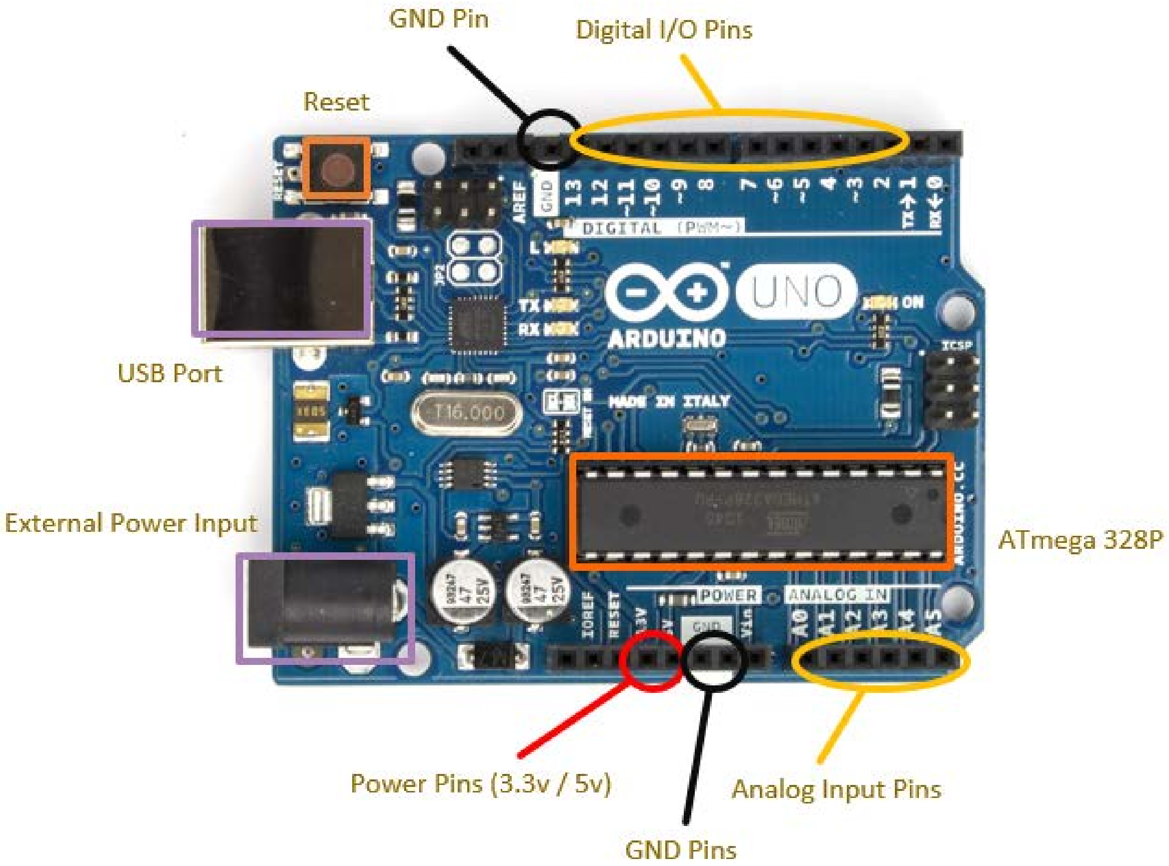


**Figure 4:** Anatomy of standard Arduino microcontroller board

Given its stability and durability, the **receiver system** uses Arduino Uno as the main computing resource for interpreting constant data inputs and communicating real-time information. Figure 5 is a diagram of Arduino Uno and Table 1 represents its hardware specification [7].

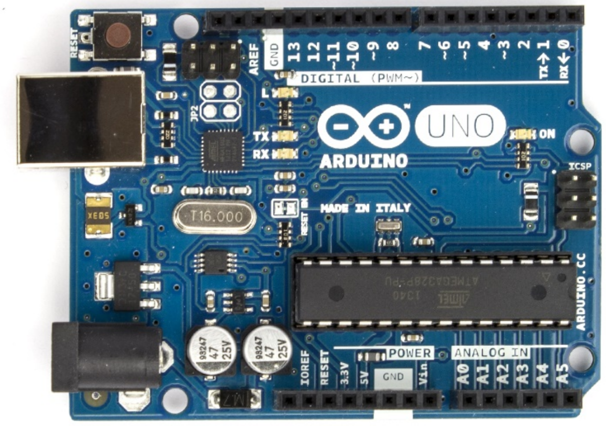


**Figure 5:** Arduino Uno

| Arduino Uno Summary | |
|---|---|
| Microcontroller | ATmega 328 |
| Physical Dimension | 2.7" x 2.3" |
| Operating Voltage | 5 V |
| Input Voltage Range | 6 – 20 V |
| Analog Input Pins | 6 |
| Digital I/O Pins | 14 |
| DC Current per I/O Pin | 40 mA |
| Max Clock Speed | 16 MHz |

**Table 1:** Arduino Uno - hardware specification

Unlike the receiver system which has a steady power source, the transmitter system acquires its power source from the solar panel with a backup LiPo battery. Thus, it is important to pick an Arduino model with small footprint and minimum power consumption. From Equation 1, power (P) is directly proportional to current (I) when voltage (V) remains the same.

$$P = I * V \tag{1}$$

This means that higher current in a circuit board normally consumes more power with constant voltage supply.

Monk's Arduino book offers a power consumption table, as shown in Table 2, for several Arduino microcontroller boards [8].

| Arduino Power Consumptions | |
|---|---|
| Uno (5V USB) | 47 mA |
| Uno (9V power supply) | 48 mA |
| Due (5V USB) | 160 mA |
| Due (9V power supply) | 70 mA |
| Pro Mini (9V power supply) | 42 mA |
| Pro Mini (5V USB) | 22 mA |
| Pro Mini (3.3V direct) | 8 mA |

**Table 2:** Power consumption comparison for selected Arduino models

Clearly, Arduino Pro Mini (3.3V) is the best microcontroller candidate for the **transmitter system**. Figure 6 is a diagram of Arduino Pro Mini and Table 3 represents its hardware specification [9].

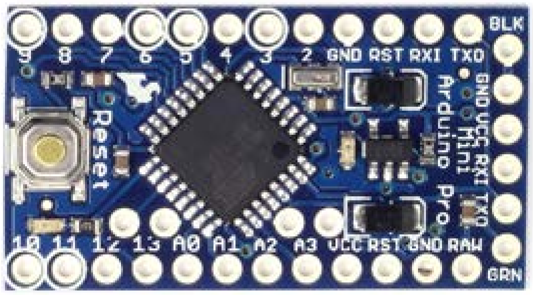

BLK
9 8 7 6 5 4 3 2 GND RST RXI TXO
Reset
Arduino Pro Mini
GND VCC RXI TXO
10 11 12 13 A0 A1 A2 A3 VCC RST GND RAW
GRN

**Figure 6:** Arduino Pro Mini

| Arduino Pro Mini (3.3 V) Summary | |
|---|---|
| Microcontroller | ATmega 328 |
| Physical Dimension | 1.3" x 0.7" |
| Operating Voltage | 3.3 V |
| Input Voltage Range | 3.35 – 12 V |
| Analog Input Pins | 8 |
| Digital I/O Pins | 14 |
| DC Current per I/O Pin | 40 mA |
| Max Clock Speed | 8 MHz |

**Table 3:** Arduino Pro Mini (3.3V) - hardware specification

### *2.2.2 Power Management Module*

Keeping constant power supply to the transmitter system is extremely critical. The power management module transfers solar power into electricity and ensures continuous system operation by using an efficient solar panel, a LiPo battery, and a Lithium-ion/LiPo charge management controller.

Since it would be difficult to frequently update the transmitter system's power supply, capacity rating must be considered when selecting a rechargeable battery for the transmitter system. Equation 2 offers a quick estimation of battery life based on battery's capacity rating and total current usage.

$$Battery\ Life\ [days] = \frac{Capacity\ Rating\ [mAh]}{Total\ Current\ [mA] * 24\ [\frac{hours}{day}]} \qquad (2)$$

From this equation, battery life improves when using a battery with higher capacity rating.

Also, Lithium Polymer batteries offers enhanced protection against overcharge and lightweight package comparing to traditional Lithium-ion batteries. Therefore, a high capacity LiPo battery (3.7V 2500mAh), as shown in Figure 7, is selected to supply electricity when solar power is absent.

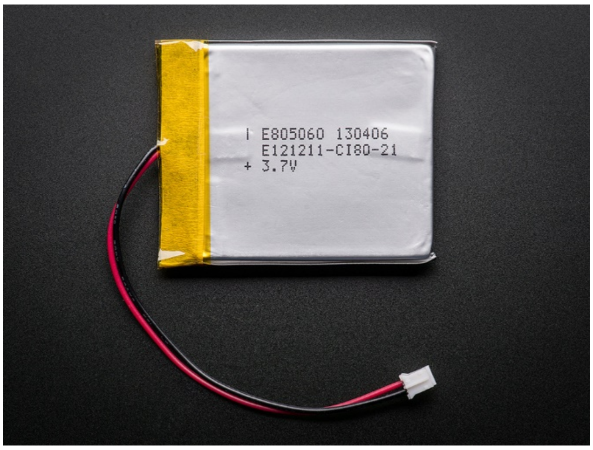


**Figure 7:** LiPo battery - 3.7V 2500mAh

Next, the power management module adopts a 6V 2W solar panel, as shown in Figure 8, as its main power source.

**Figure 8:** Solar panel - 6V 2W

A power regulation circuit needs to manage the charging/discharging process once main power source (solar panel) and backup power supply (LiPo battery) are determined. Due to its ease of use and bountiful application examples, the MCP73833 IC package is adopted as the power management controller. Based on MCP73833 datasheet's application example and online discussion, Figure 9 displays schematic design of the power regulation circuit [10] [11].

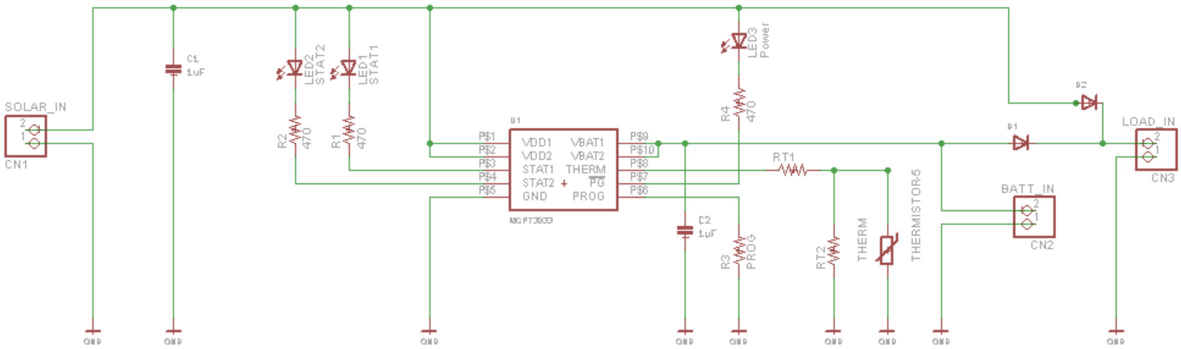


**Figure 9:** MCP7833-based power management circuit

From the datasheet, a programming resistor ($R_{PROG}$) can be applied to set the charge current by using Equation 3 [10].

$$I_{REG}\ [mA] = \frac{1000\ [V]}{R_{PROG}\ [k\Omega]} \tag{3}$$

Additionally, Equation 4 and Equation 5 set the charge temperature range by using two resistors and a thermistor [10].

$$24\ [k\Omega] = R_{T1} + \frac{R_{T2} * R_{COLD}}{R_{T2} + R_{COLD}} \tag{4}$$

$$5\ [k\Omega] = R_{T1} + \frac{R_{T2} * R_{HOT}}{R_{T2} + R_{HOT}} \tag{5}$$

Where $R_{T1}$ and $R_{T2}$ are fixed resistance values and $R_{COLD}$ and $R_{HOT}$ are thermistor resistance values of desired temperature range.

For this power management module, the resistance values are as the following: $R_{PROG} = 2.2\ k\Omega$ for matching a charge current of 455 mA; $R_{T1} = 2.2\ k\Omega$, $R_{T2} = 100\ k\Omega$ for a 10 $k\Omega$ 25C NTC thermistor as suggested by the datasheet ($R_{T1} = 1.54\ k\Omega$, $R_{T2} = 69.8\ k\Omega$) [10].

### *2.2.3 Water Sensor Module*

Water sensor module is invented based on the simple fact of water's electrical conductivity. Most types of water contain aquatic ions that expedite electron transfer. Figure 10 shows design concept of the water sensor module. In here, water probe is open circuit when water level is below the sensor, but it becomes a closed circuit and signals the transmitter system once water level reaches the probe.

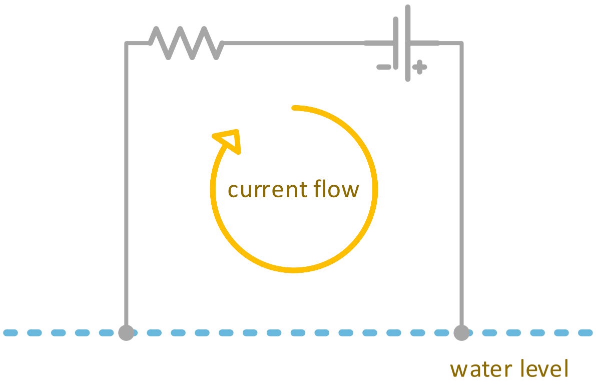


**Figure 10:** Design concept for water sensor module

Figure 11 is an example of commercial moisture probes.

**Figure 11:** Commercial moisture sensor

Nevertheless, heavy rainfall tends to dilute ion concentration in fresh water. This reduces or even eliminates electrical conductivity in water (i.e., distilled water does not conduct electricity). To combat ion loss in water during heavy rains, a modified water level sensor includes a small container with salts (e.g., table salt) to increase electrical conductivity. Figure 12 represents the revised sensor design diagram.

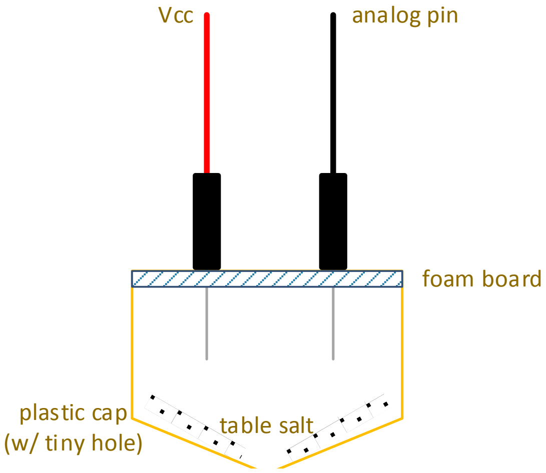


**Figure 12:** Customized water sensor module

### *2.2.4 RF Transceiver Module*

As a best seller in the largest online store, nRF24L01+ transceiver module is known for its ease of use, small power usage, and extremely low cost. Figure 13 shows actual package of the nRF24L01+ 2.4GHz wireless transceiver. Table 4 portrays major characteristics of the transceiver module based on the product datasheet [12].

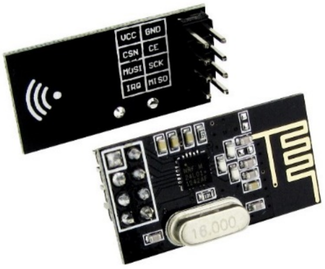


**Figure 13:** nRF24L01+ 2.4GHz Wireless Transceiver

| nRF24L01+ Transceiver Summary | |
|---|---|
| Operating Frequency | 2.4 – 2.5 GHz |
| Power Consumption (Tx) | 7.0 – 11.3 mA |
| Power Consumption (Rx) | 8.9 – 13.5 mA |
| Supply Voltage | 1.9 – 3.6 V |
| Operating Temperature | -40 – 85 °C |

**Table 4:** nRF24L01+ Transceiver - hardware specification

### 2.2.5 *Information Display Module*

A monitor display module is created to deliver real-time information of water level and transmitter power status. The following information is available on the status dashboard:

- Transmitter (Tx) power status – Green LED
- Water level 1 (L1) – Yellow LED
- Water level 2 (L2) – Amber LED
- Water level 3 (L3) – Red LED
- Pre-flood warning – White LED

Figure 14 displays schematic design of the information display module.

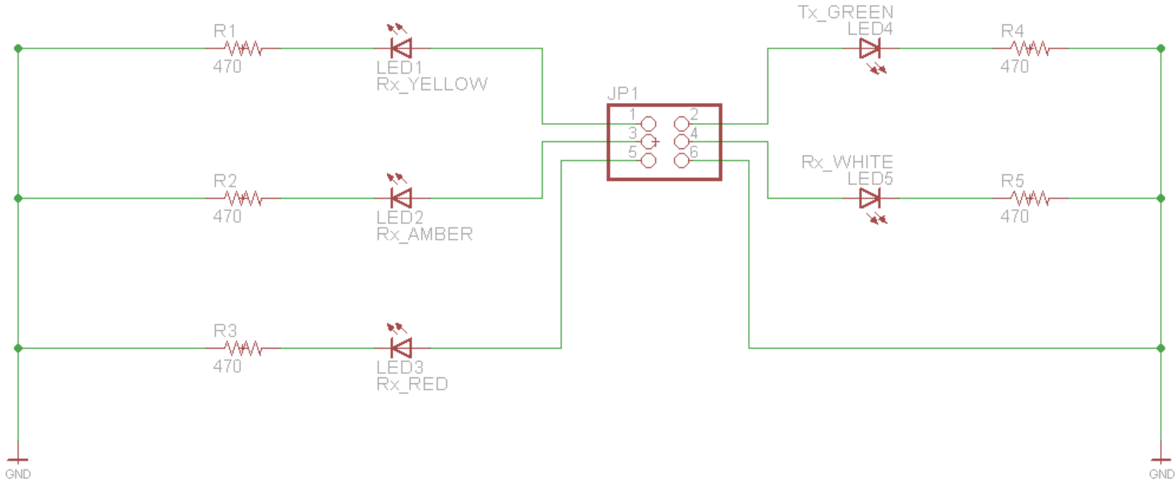


**Figure 14:** Information display circuit

### 2.2.6 *High Water Siren Module*

Besides visual status indicator, a high pitch siren is needed to add acoustic warning when the receiver system detects flood risk. Figure 15 is a buzzer that offers low power consumption and simple configuration.

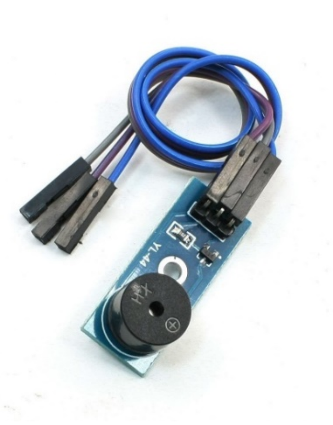


**Figure 15:** 9012 Transistor Driver Buzzer

## 2.3 Software Design

Both Arduino microcontroller boards were programmed using the Arduino integrated development environment (IDE). Figure 16 illustrates a regular Arduino IDE window.

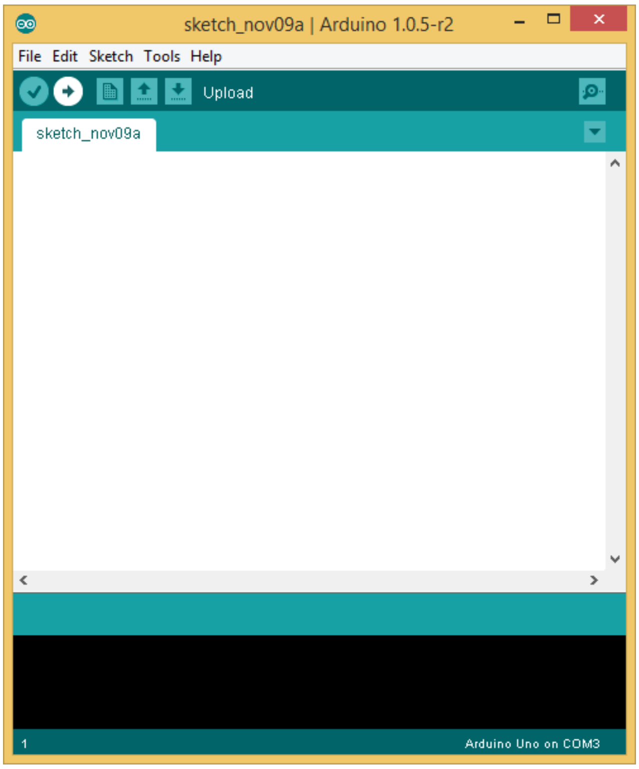


**Figure 16:** Arduino IDE

Based on the C language, the Arduino IDE includes a variety of software libraries such as Ethernet and Wi-Fi. Additional software libraries can be easily imported as well.

In this project, the software design strategy for the transmitter system is minimizing power consumption by

- Lowering the clock speed [8],
- Storing individual data into an array for easy transmission, and
- Forwarding raw data to the receiver system without excessive processing [13].

However the receiver system's software design strategy uses the opposite approach. It implements several logical loops to ensure proper event notifications. The following subsections describes programming for the transmitter system and the receiver system.

### *2.3.1 Transmitter System*

Figure 17 is the logic diagram for programming the transmitter system.

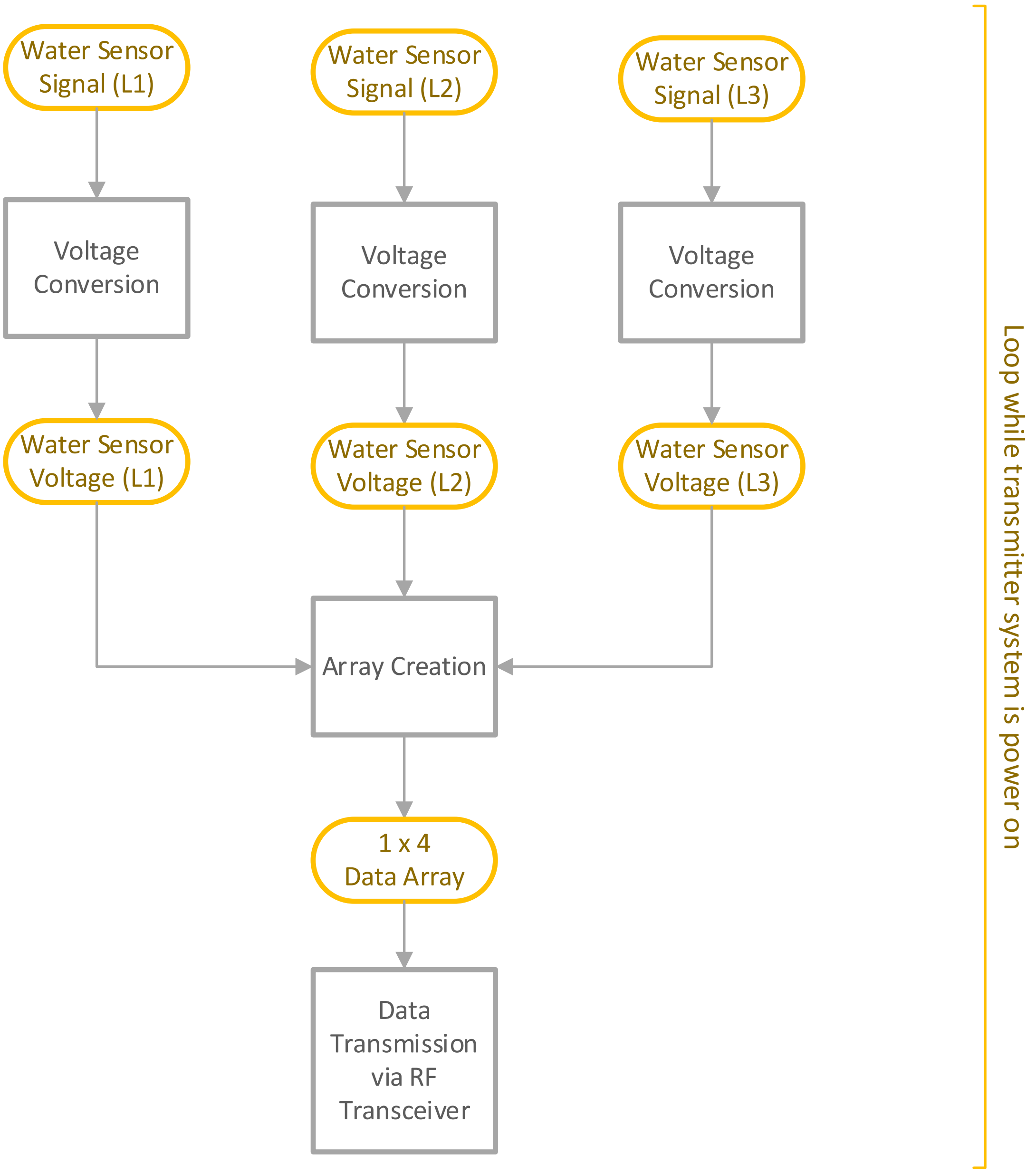


**Figure 17:** Logic diagram for coding transmitter system

With reduced clock speed to save power, the transmitter program is designed to actively listen to all three water sensors and to constantly transmit sensor data to the receiver system. The complete code can be found in Appendix A.

### *2.3.2 Receiver System*

Figure 18 is the logic diagram for programming the receiver system.

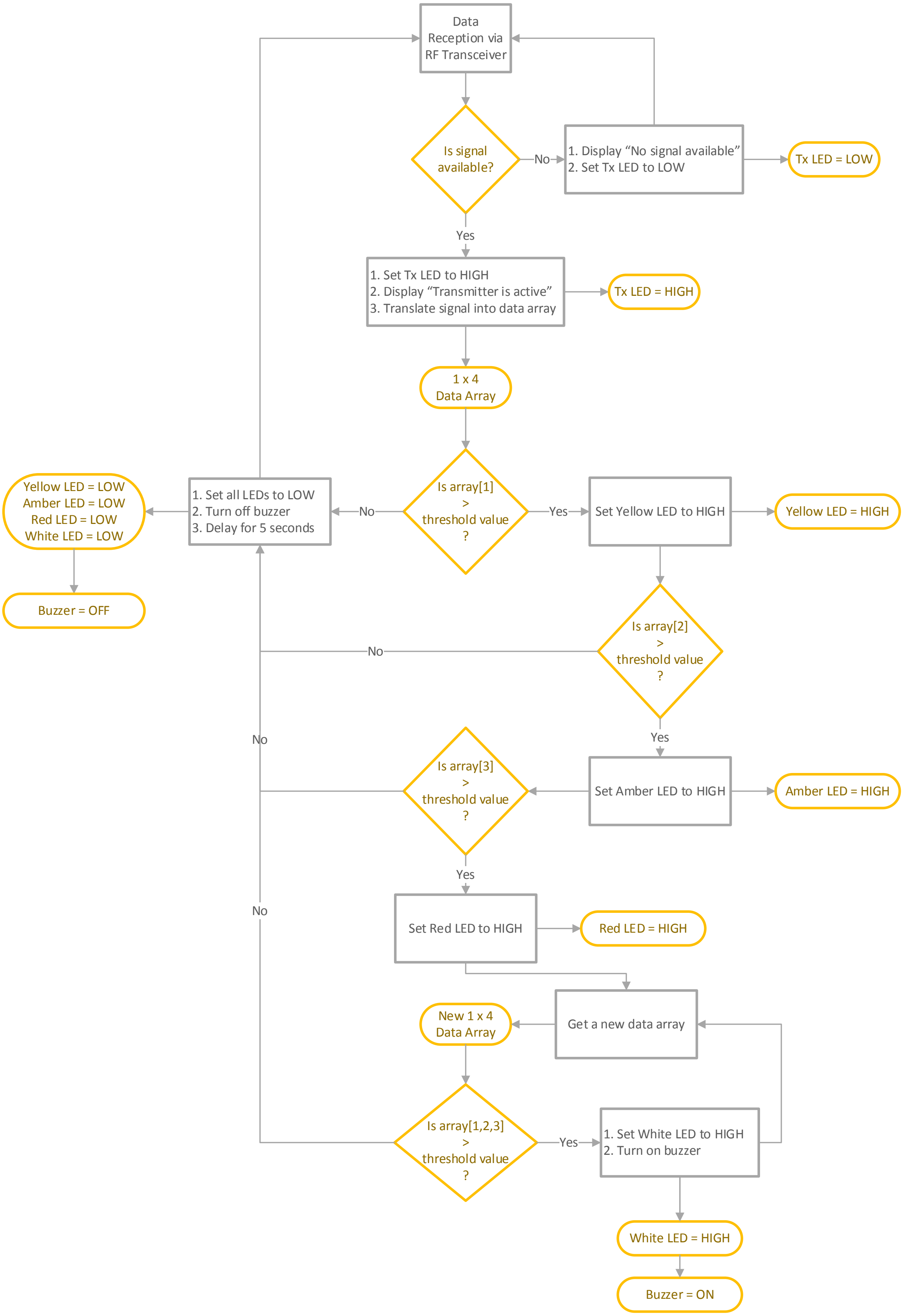


**Figure 18:** Logic diagram for coding receiver system

The receiver program is intended to constantly fetch raw data from the transmitter and to apply comprehensive logics to identify any indication of rising water. The complete code can be found in Appendix B.

# 3 Implementation

Once the system design and materials selection are complete, the wireless high water detection system can be easily assembled. The following subsections introduce components acquisition and system integration.

## 3.1 PCB Fabrication

Two circuit designs, power management module and information display module, must translate into printed circuit board (PCB) layouts prior system implementation. Figure 19 is PCB layout of the power management circuit. In here, two 1N4148 switching diodes are exerted in place of 5mm load LEDs to reduce forward voltage drop.

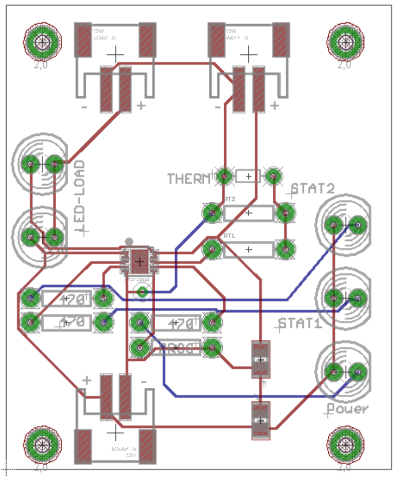


**Figure 19:** PCB layout of the power management circuit

Figure 20 is PCB layout of the information display circuit.

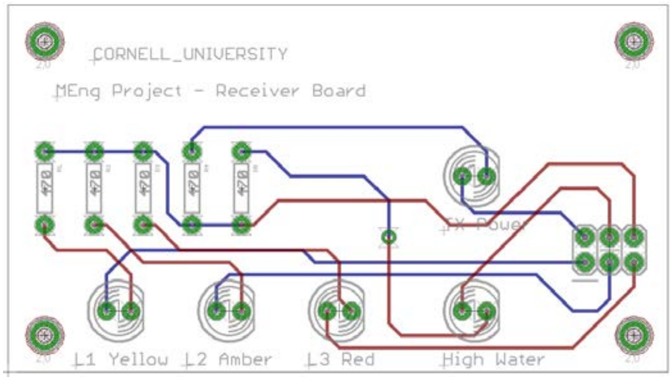
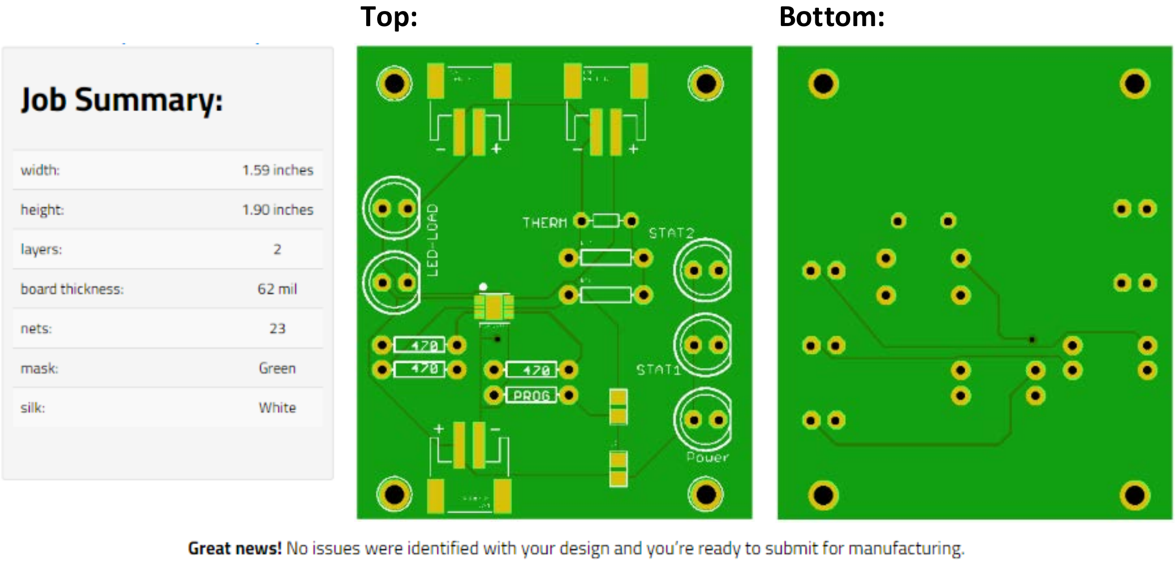


**Figure 20:** PCB layout of the information display circuit

After drafting the PCB layouts, both PCB drawings are submitted for fabrication. Figure 21 and Figure 22 exhibit the pre-fabrication checks offered by a third-party manufacturing company.

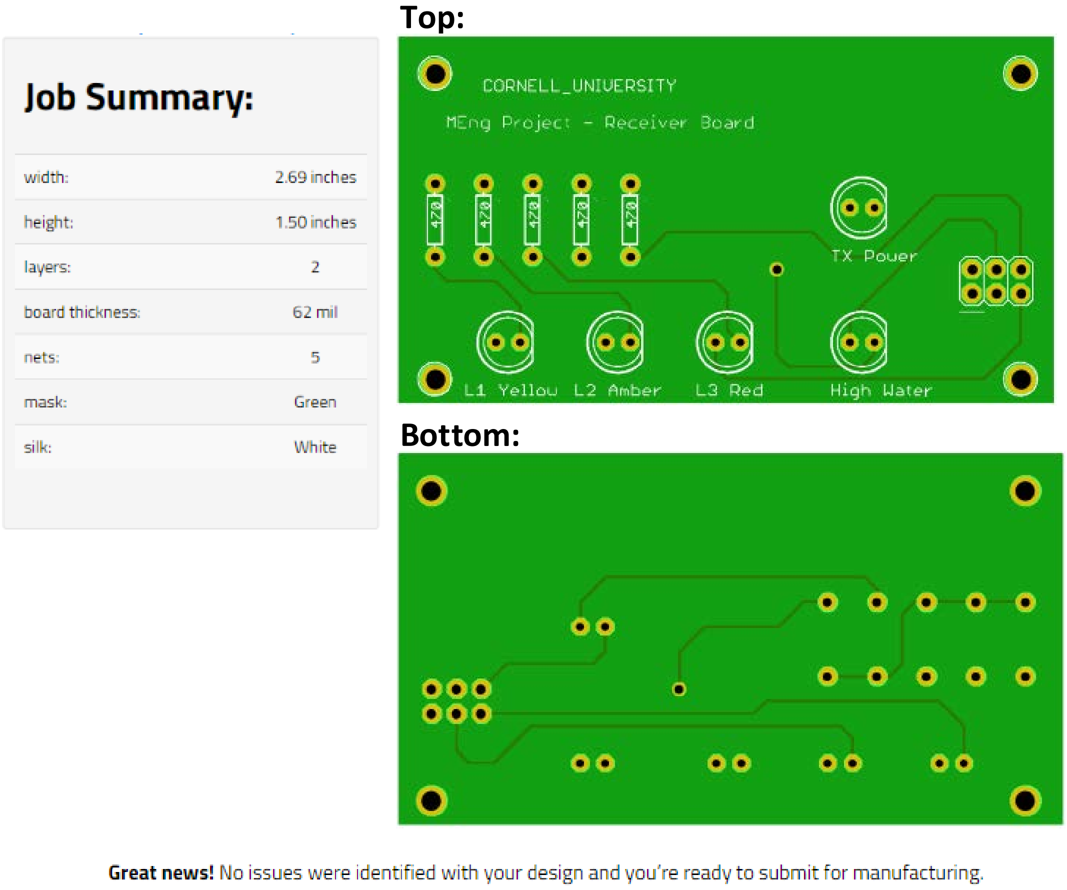


## Job Summary:

| | |
|---|---|
| width: | 1.59 inches |
| height: | 1.90 inches |
| layers: | 2 |
| board thickness: | 62 mil |
| nets: | 23 |
| mask: | Green |
| silk: | White |

**Great news!** No issues were identified with your design and you're ready to submit for manufacturing.

**Figure 21:** InstantDFM check for the power management PCB



## Job Summary:

| | |
|---|---|
| width: | 2.69 inches |
| height: | 1.50 inches |
| layers: | 2 |
| board thickness: | 62 mil |
| nets: | 5 |
| mask: | Green |
| silk: | White |

**Great news!** No issues were identified with your design and you're ready to submit for manufacturing.

**Figure 22:** InstantDFM check for the information display PCB

Most circuit elements are through-hole components for easy soldering.

## 3.2 Transmitter System Integration

Figure 23 exemplifies wire connections among the transmitter system components.

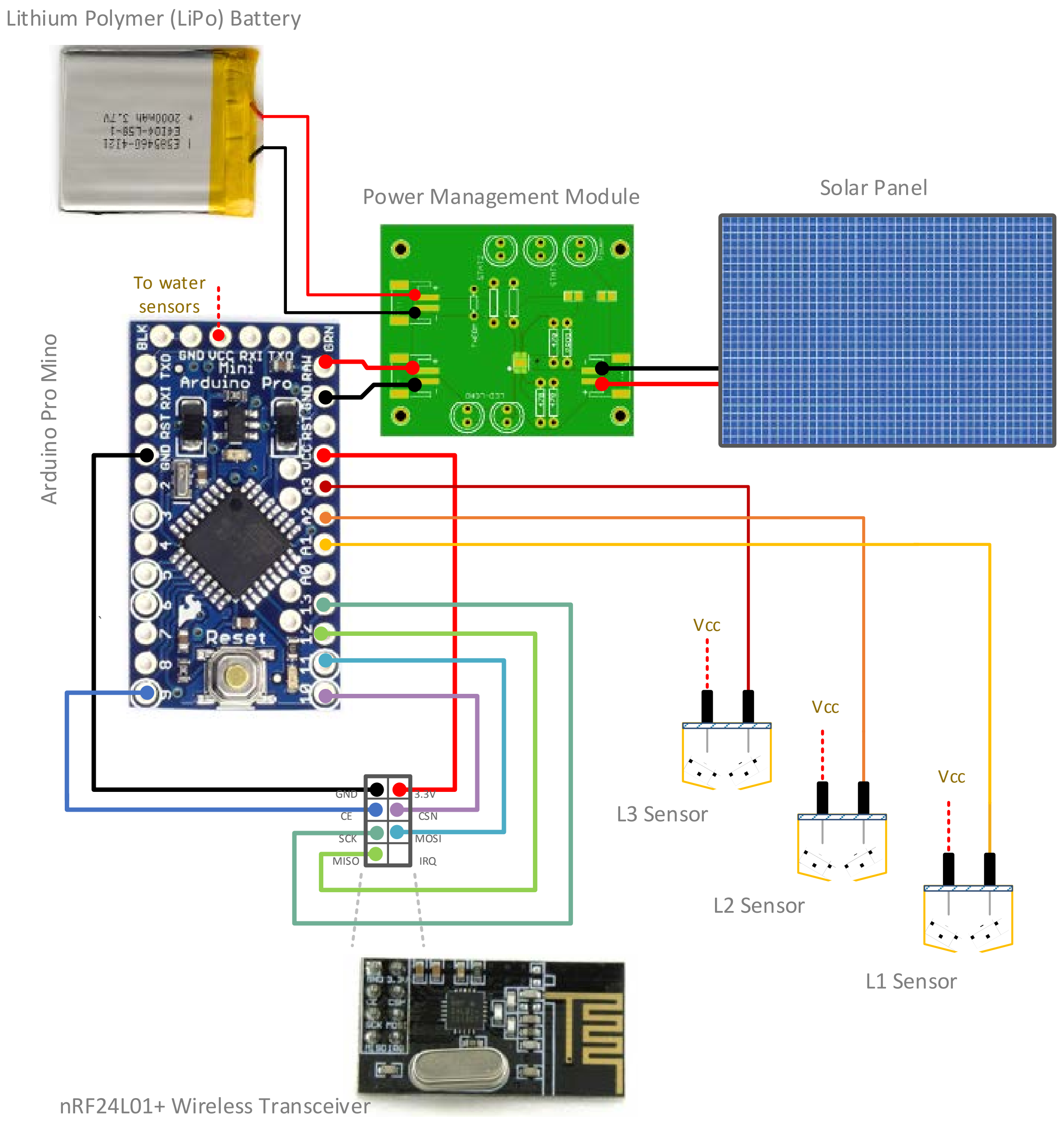


**Figure 23:** Implementation diagram for the transmitter system

The transmitter system can be integrated with the following steps.

1. Connect cable wires for the nRF24L01+ wireless transceiver module
2. Connect wires for the water level sensors
3. Connect cable wires for the power management module, including solar panel, LiPo battery, and power regulation board
4. Upload the transmitter Arduino sketch (program)

## 3.3 Receiver System Integration

Figure 24 exemplifies wire connections among the receiver system components.

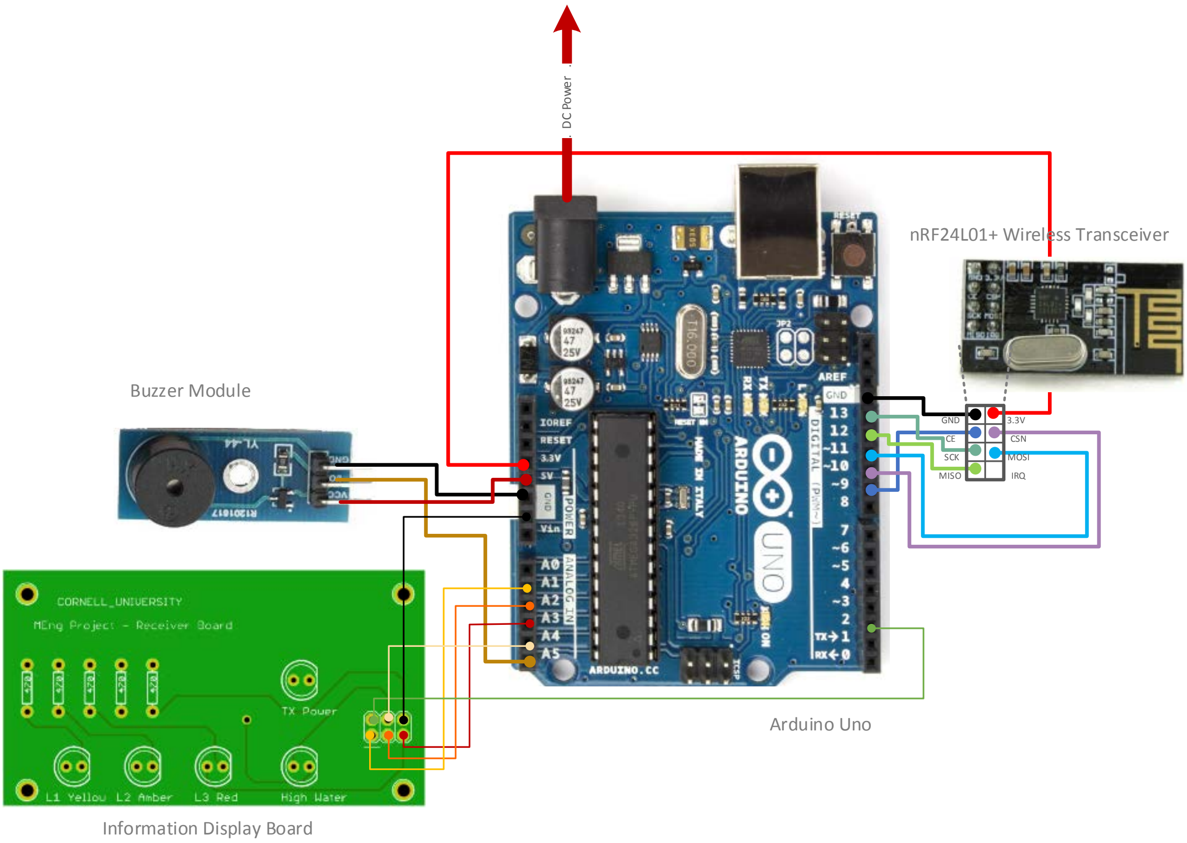


**Figure 24:** Implementation diagram for the receiver system

The receiver system can be integrated with the following steps.

1. Connect cable wires for the nRF24L01+ wireless transceiver module
2. Connect cable wires for the information display board
3. Connect cable wires for the buzzer module
4. Apply DC power source (e.g., USB cable) to turn on the Arduino board
5. Upload the receiver Arduino sketch (program)

## 3.4 Alternative Receiver System Implementation Method

Alternatively, the receiver system modules can be further consolidated with a tailored Arduino shield. Arduino shield is a customizable circuit board that mounts on top of the Arduino microcontroller board. It hosts multiple electronic components on the same board which reduces external wire counts and implementation complexity for Arduino-based projects. The schematic design and PCB layout of the receiver Arduino shield are shown in Figure 25 and Figure 26.

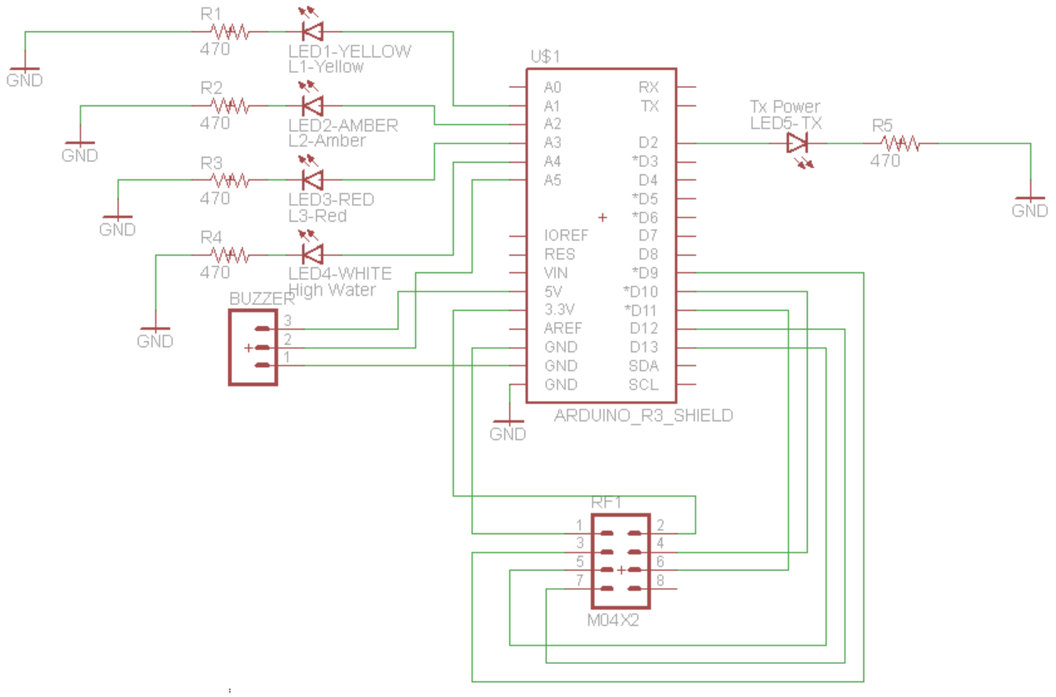


**Figure 25:** Schematic design of the receiver Arduino shield

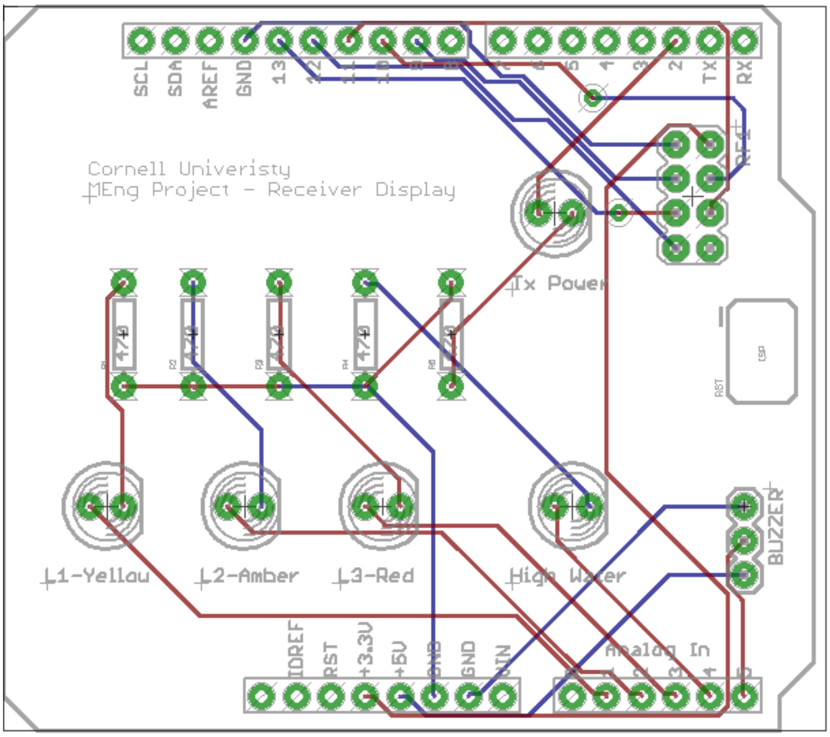


**Figure 26:** PCB layout of the receiver Arduino shield

By adopting this shield, information display, wireless transceiver, and buzzer are now combined into a single board.

## 3.5 Populated PCBs

After soldering necessary electronic parts, Figure 27, Figure 28, and Figure 29 illustrate the fully integrated PCBs and their pre-populated boards.

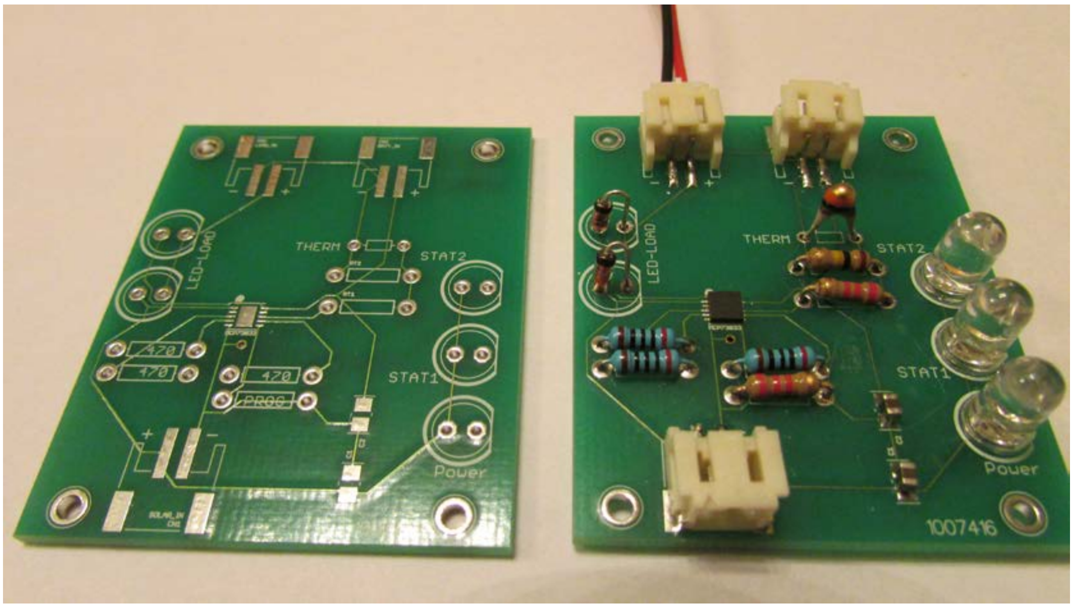


**Figure 27:** Power management boards

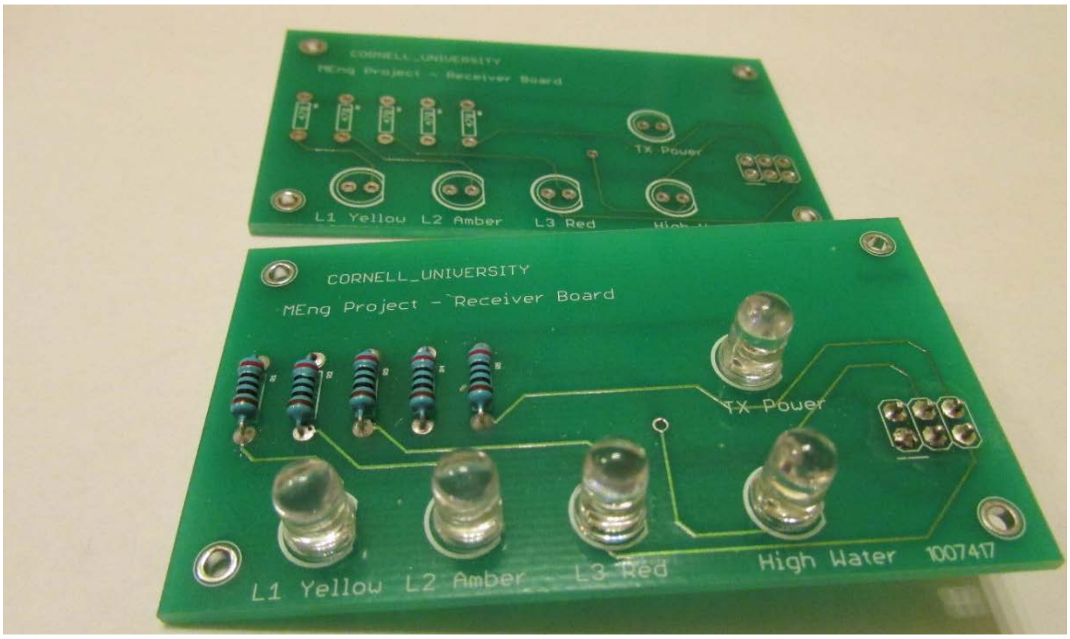


**Figure 28:** Information display boards

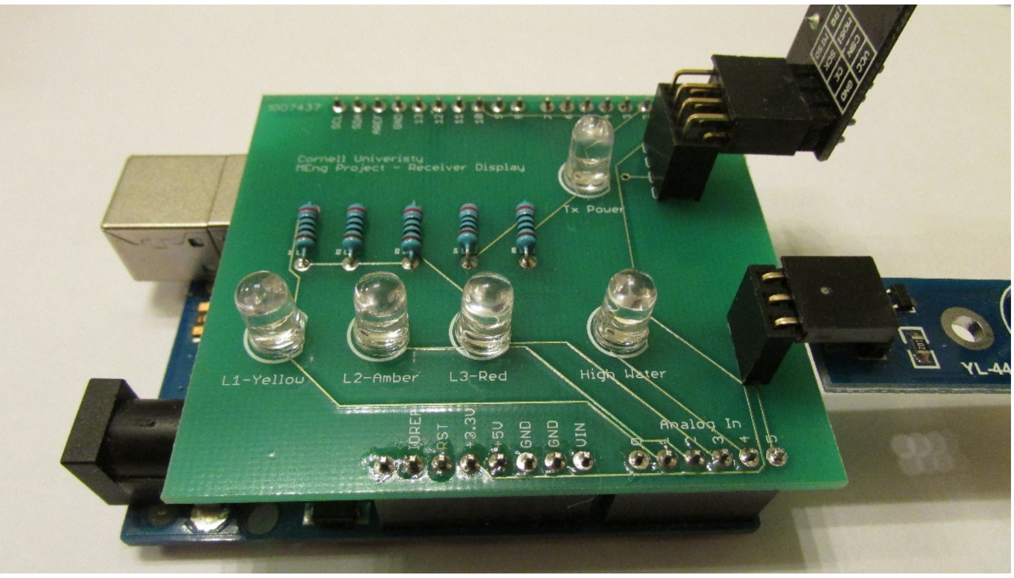


**Figure 29:** Arduino shield

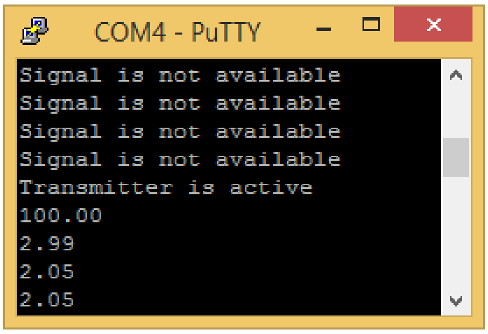
# 4 Testing and Results

System testing divides into two parts: proof-of-concept evaluation and field testing. Both sections are carefully examined using test cases outlined in Table 5.

| # | Transmitter System Events | Receiver System Actions | Results |
|---|---|---|---|
| 1 | Transmitter - power off | Green LED = OFF | |
| 2 | Transmitter - power on | Green LED = ON | |
| 3 | L1 sensor triggered | Yellow LED = ON | |
| 4 | L1,L2 sensors triggered | Amber LED = ON | |
| 5 | L1, L2, L3 sensors triggered | RED LED = ON | |
| 6 | All sensors triggered for 30+ seconds | White LED, buzzer = ON | |
| 7 | All sensors - turn off | All LEDs except Green LED = OFF | |

**Table 5:** Test cases for evaluating system performance

To simplify software debugging, each program action is also described in plain texts and projected on a serial monitor. Figure 30 is an example of the receiver actions captured on a serial display.



```
Signal is not available
Signal is not available
Signal is not available
Signal is not available
Transmitter is active
100.00
2.99
2.05
2.05
```

**Figure 30:** Serial monitor via PuTTY

## 4.1 Prototype Testing

An Arduino-based prototype is created to prefect software programs and to establish performance expectations. Both beta systems, as shown in Figure 31 and Figure 32, are built using Arduino Uno (Revision 3) for quick setup.

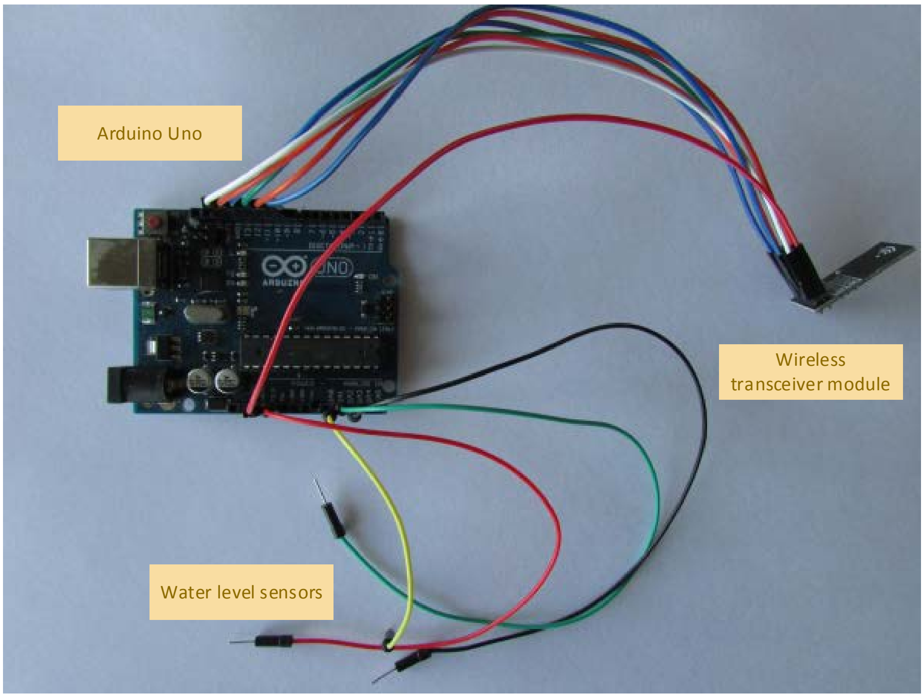


**Figure 31:** Transmitter system prototype

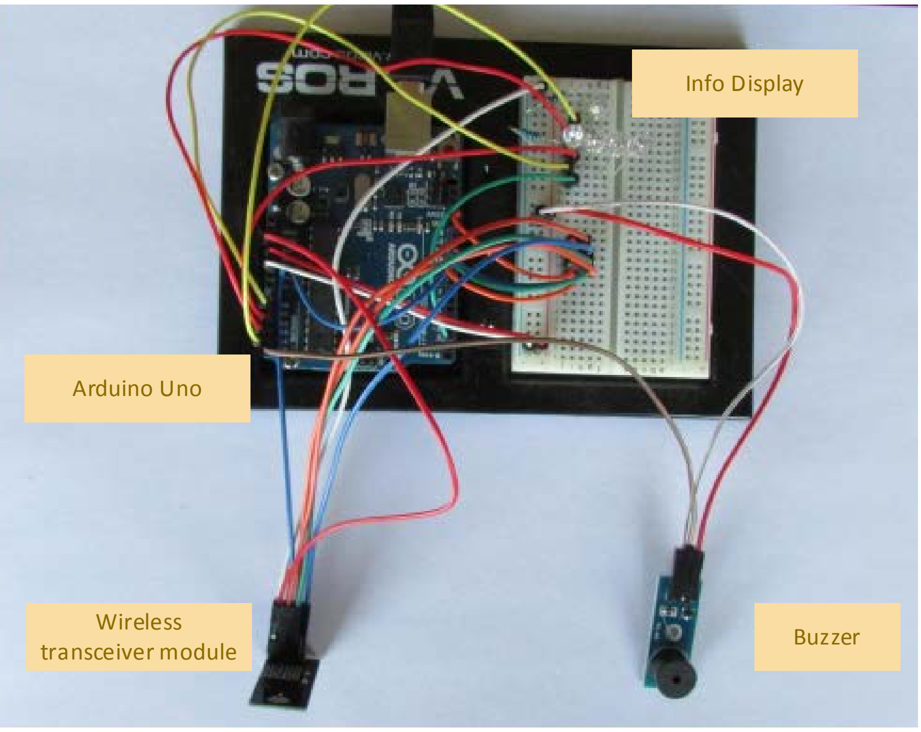


**Figure 32:** Receiver system prototype

Moreover, some water sample is collected from a local creek to carry out the experiment.

Prior executing the test cases, the transmitter system is placed approximately 30 meters away from the receiver system with couple barriers in between (i.e., non-line of sight) and both systems are monitored via serial displays. Then the prototype testing proceeds by going over the predefined test cases. After numerous trials, Table 6 shows the ultimate results for this testing.

| # | Transmitter System Events | Receiver System Actions | Results |
|---|---|---|---|
| 1 | Transmitter - power off | Green LED = OFF | Success! |
| 2 | Transmitter - power on | Green LED = ON | Success! |
| 3 | L1 sensor triggered | Yellow LED = ON | Success! |
| 4 | L1,L2 sensors triggered | Amber LED = ON | Success! |
| 5 | L1, L2, L3 sensors triggered | RED LED = ON | Success! |
| 6 | All sensors triggered for 30+ seconds | White LED, buzzer = ON | Success! |
| 7 | All sensors - turn off | All LEDs except Green LED = OFF | Success! |

**Table 6:** Results for prototype testing

Many valuable lessons are learned and minor tweaks are made to improve the final product.

## 4.2 Field Testing

The field testing is conducted at a remote trail with creek nearby to simulate the real setting. Figure 33 is the test location.

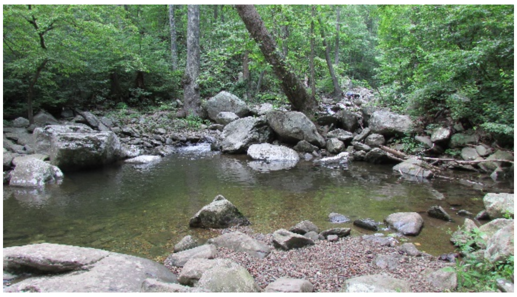


**Figure 33:** Field test location

Same as prototype testing, the transmitter system is placed about 30 meters away from the receiver system with couple trees in between. Then the test cases are performed and their results are noted in Table 7.

| # | Transmitter System Events | Receiver System Actions | Results |
|---|---|---|---|
| 1 | Transmitter - power off | Green LED = OFF | Success! |
| 2 | Transmitter - power on | Green LED = ON | Success! |
| 3 | L1 sensor triggered | Yellow LED = ON | Success! |
| 4 | L1,L2 sensors triggered | Amber LED = ON | Success! |
| 5 | L1, L2, L3 sensors triggered | RED LED = ON | Success! |
| 6 | All sensors triggered for 30+ seconds | White LED, buzzer = ON | Success! |
| 7 | All sensors - turn off | All LEDs except Green LED = OFF | Success! |

**Table 7:** Results for field testing

# 5 Conclusion

The wireless high water detection system is built to identify rising water levels and to warn any potential flood risk. Using solar panel and power management module, the transmitter system is able to serve for a long period of time with minimum maintenance. The receiver system's straightforward dashboard design gives user a fast update of current water level. This flood detection model is suitable for all outdoor and indoor applications, especially for locations without cellphone signal coverage.

## 5.1 Future Improvement

The system can be further enhanced by reducing PCB size, lowering overall cost, and measuring water level at multiple sites.

- Reducing PCB size – The power management board size can easily shrink up to 30% by replacing through-hole components with surface mount device (SMD) elements.
- Lowering overall cost – There are several ways to lower costs. For example, Arduino boards can be substituted with equivalent microcontrollers. Another method is to exploit generic electronic parts instead of using branded components.
- Measuring water level at multiple sites – It is possible to monitor water levels at more than one location. To achieve this, a sensor network design and extensive transceiver testing are needed. ZigBee module is a good candidate for this application.

# 6 Acknowledgement

I would like to thank Professor Bruce Land for his technical guidance and instruction during this process. His insights for selecting electronic components and setting threshold values allow me to avoid many potential hassles. Dr. Land always responds my questions with great enthusiasm and precise directions. I truly appreciate his dedication in helping me to move forward.

# 7 References

[1] Federal Emergency Management Agency. (2014). “The Cost of Flooding.” *Federal Emergency Management Agency*. [Online]. Available: https://www.floodsmart.gov

[2] “Students' Project Makes Low-Lying Campgrounds Safer from Flooding.” (2011, Jun. 15). *The University of Arkansas Newswire*. [Online]. Available: http://newswire.uark.edu

[3] National Oceanic and Atmospheric Administration, National Weather Service, *Flood Warning Systems Manual*. Washington, DC: NWS, 2012. [Online]. Available: http://www.nws.noaa.gov

[4] InspectUSA. (2014). *Battery Operated Flood Alarm with Remote Sensor*. [Online]. Available: http://inspectusa.com

[5] Specialty Alarms & Security Products. (2005-2013). *WaterBug 350 Water Detection Sensor*. [Online] Available: http://www.specialtyalarms.com

[6] PumpAlarm. (2014). *Cellular Water and Power Alarm*. [Online]. Available: http://store.pumpalarm.com

[7] Arduino. (2014). “Arduino Uno.” *Arduino*. [Online]. Available: http://arduino.cc

[8] S. Monk, *Programming Arduino Next Steps: Going Further with Sketches*. New York: McGraw-Hill, 2013, pp. 85-91.

[9] Arduino. (2014). “Arduino Pro Mini.” *Arduino*. [Online]. Available: http://arduino.cc

[10] Microchip. (2006). “MCP73833/4 Stand-Alone Linear Li-Ion/Li-Polymer Charge Management Controller Datasheet.” *Microchip*. [Online]. Available: http://ww1.microchip.com

[11] “Li-po Charging Circuit Question (Switching Between Battery and Power Supply).” (2014, Apr. 2). *StackExchange*. [Online]. Available: http://electronics.stackexchange.com

[12] Nordic Semiconductor. (2007). “nRF24L01 Single Chip 2.4 GHz Transceiver Product Specification.” *Nordic Semiconductor*. [Online]. Available: http://www.nordicsemi.com

[13] T. King. (2014, Aug. 28). “NRF20L01-2.4GHz-HowTo.” *Arduino-Info*. [Online]. Available: http://arduino-info.wikispaces.com

[Figure 4] Arduino. (2014). “Arduino Uno.” *Arduino.* [Online]. Available: http://arduino.cc

[Figure 5] Arduino. (2014). “Arduino Uno.” *Arduino.* [Online]. Available: http://arduino.cc

[Figure 6] Arduino. (2014). “Arduino Pro Mini.” *Arduino.* [Online]. Available: http://arduino.cc

[Figure 7] Adafruit. (2014). “Lithium Ion Polymer Battery.” *Adafruit.* [Online]. Available: http://www.adafruit.com/

[Figure 8] Adafruit. (2014). “Medium 6V 2W Solar Panel.” *Adafruit.* [Online]. Available: http://www.adafruit.com/

[Figure 11] Emartee. (2014). “High Sensitivity Moisture Sensor.” *Emartee.* [Online]. Available: http://www.emartee.com/

[Figure 13] Addicore. (2014). “2pcs nRF24L01+ 2.4GHz Wireless Transceiver Arduino Compatible.” *Amazon.com.* [Online]. Available: http://www.amazon.com/

[Figure 15] Gino. (2014). “DC 3V-5V 9012 Transistor Driver Buzzer Module with 3 Wires.*”* *Amazon.com.* [Online]. Available: http://www.amazon.com/

# 8 Appendix

## 8.1 Appendix A - Arduino Sketch for the Transmitter System

```
/* Cornell University - MEng Project */
/* Transmitter System Codes          */
/* Key Reference: Arduino playground - nRF24L01 How-To and its example codes [13]*/
/* Note: most nRF24L01-related codes are referenced from the above source */
/* Note: Monk's Programming Arduino Next Steps book offers a great way to reduce clock
speed with the Prescaler library [8]. */


/* import libraries */
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <Prescaler.h>


/* establish RF communication [13] */
float signal[4];
RF24 radio(9,10);
const uint64_t pipe = 0xE8E8F0F0E1LL;


/* declare variables */
int normalPin = A0;
int yellowPin = A1;
int amberPin = A2;
int redPin = A3;


/* initialize system setup */
void setup(){
  setClockPrescaler(CLOCK_PRESCALER_256); // set clock rate to 62.5 kHz to save energy
  Serial.begin(9600);
  Serial.println("Cornell MENG Design Project");
  Serial.println("Wireless Flood Detection System - Maintenance Mode");
  radio.begin();
  radio.openWritingPipe(pipe);
}


/* constantly obtain water levels and send to receiver */
void loop(){
  int normalInput = analogRead(normalPin);
  int yellowInput = analogRead(yellowPin);
  int amberInput = analogRead(amberPin);
  int redInput = analogRead(redPin);
  float yellowVoltage = yellowInput / 204.6; // voltage conversion
  float amberVoltage = amberInput / 204.6;   // voltage conversion
  float redVoltage = redInput / 204.6;       // voltage conversion

  signal[0] = 100.00;
  signal[1] = yellowVoltage;
  signal[2] = amberVoltage;
  signal[3] = redVoltage;

  Serial.println("");
  Serial.println("");
  Serial.println("yellow voltage:");
  Serial.println(signal[1]);
  Serial.println("amber voltage:");
  Serial.println(signal[2]);
  Serial.println("red voltage:");
  Serial.println(signal[3]);
  Serial.println("");
  radio.write(signal, sizeof(signal));
}
```

## 8.2 Appendix B - Arduino Sketch for the Receiver System

```
/* Cornell University - MEng Project */
/* Receiver System Codes             */
/* Key Reference: Arduino playground - nRF24L01 How-To and its example codes [13]*/
/* Note: most nRF24L01-related codes are referenced from the above source */

/* import libraries */
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

/* establish RF communication [13] */
float signal[4];
RF24 radio(9,10);
const uint64_t pipe = 0xE8E8F0F0E1LL;

/* declare variables */
int greenLED = 2;
int yellowLED = 3;
int amberLED = 4;
int redLED = 5;
int whiteLED = 6;

/* initial system setup */
void setup(){
  Serial.begin(9600);
  radio.begin();
  radio.openReadingPipe(1,pipe);
  radio.startListening();

  pinMode(greenLED, OUTPUT);
  pinMode(yellowLED, OUTPUT);
  pinMode(amberLED, OUTPUT);
  pinMode(redLED, OUTPUT);
  pinMode(whiteLED, OUTPUT);
}

void loop(){
  if (radio.available()){
    digitalWrite(greenLED, HIGH);
    Serial.println("Transmitter is active");
    // nRF24L01-related codes are referenced from [13]
    bool done = false;
    while (!done){
      done = radio.read(signal, sizeof(signal));
      for(int i=0; i<=3; i++){
         Serial.println(signal[i]);
      }
      /* yellow alert */
      if (signal[1] > 4.9) {
         analogWrite(A1,255);
         analogWrite(A2,0);
         analogWrite(A3,0);
         analogWrite(A4,0);
         noTone(A5);
         Serial.println("Yellow Alert!!");

        /* amber alert */
        if ((signal[1] > 4.9)&&(signal[2] > 4.9)) {
         analogWrite(A1,0);
         analogWrite(A2,255);
         analogWrite(A3,0);
         analogWrite(A4,0);
         noTone(A5);
         Serial.println("Amber Alert!!");
          /* red alert */
          if ((signal[1] > 4.9)&&(signal[2] > 4.9)&&(signal[3] > 4.9)) {
```

```
     analogWrite(A1,0);
     analogWrite(A2,0);
     analogWrite(A3,255);
     analogWrite(A4,0);
     noTone(A5);
     Serial.println("Red Alert!!");

    delay(30000);   // delay 30 seconds

    delay(1000);    // delay 1 second

    done = radio.read(signal, sizeof(signal));
    for(int i=0; i<=3; i++){
       Serial.println(signal[i]);
    }
    /* flood warning */
    while((signal[1] > 4.9)&&(signal[2] > 4.9)&&(signal[3] > 4.9)){
     analogWrite(A1,0);
     analogWrite(A2,0);
     analogWrite(A3,255);
     analogWrite(A4,255);
     tone(A5,500);
     Serial.println("Flood warning!!");
     delay(5000);    // delay 5 seconds

     done = radio.read(signal, sizeof(signal));
     for(int i=0; i<=3; i++){
       Serial.println(signal[i]);
    }

    }

  }
}
```

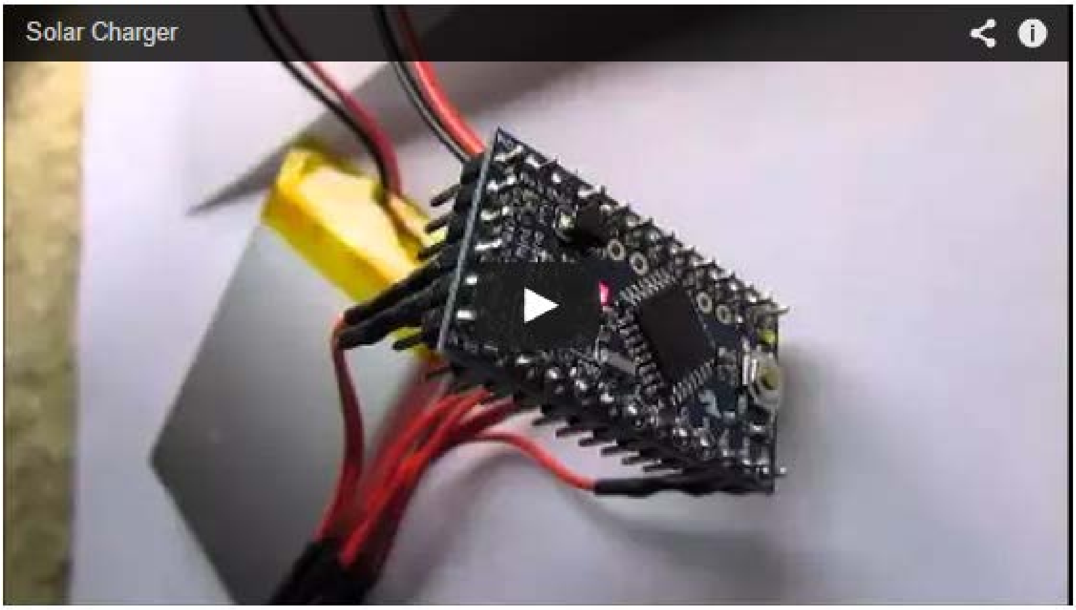
```
      }
      /* system reset */
      else if ((signal[1] < 4.9)&&(signal[2] < 4.9)&&(signal[3] < 4.9)) {
          analogWrite(A1,0);
          analogWrite(A2,0);
          analogWrite(A3,0);
          analogWrite(A4,0);
          noTone(A5);
      }

      delay(5000);    // delay 5 seconds
    }
  }
  else{
      Serial.println("Signal is not available");
      digitalWrite(greenLED, LOW);

  }
}
```

## 8.3 Appendix C - Estimated Project Cost

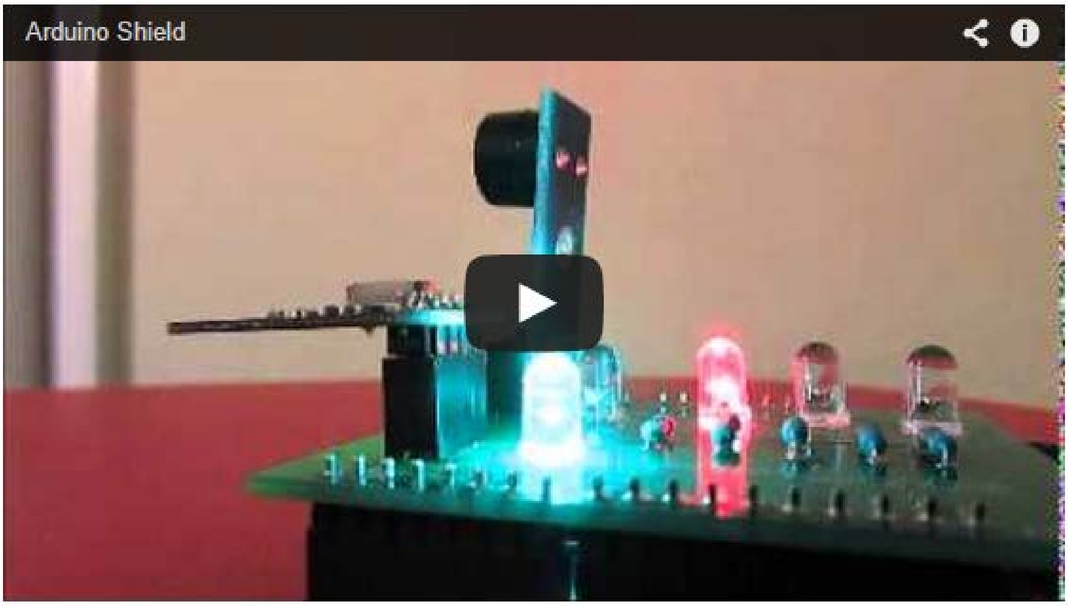
| Item | Unit Cost | Quantity | Subtotal |
|---|---|---|---|
| Arduino Uno | $14.45 | 1 | $14.45 |
| Arduino Pro Mini | $3.88 | 1 | $3.88 |
| nRF24L01 module | $2.90 | 2 | $5.80 |
| Buzzer | $3.35 | 1 | $3.35 |
| Assorted LEDs + resistors (80) | $4.01 | 1 | $4.01 |
| MCP73833 IC package | $1.02 | 1 | $1.02 |
| 2500mAh LiPo battery | $14.95 | 1 | $14.95 |
| 6W solar panel | $8.32 | 1 | $8.32 |
| Jumper wires + other costs | $4.00 | 1 | $4.00 |
| | | **Estimated Total Cost:** | $59.78[1] |

[1] PCB fabrication cost is not included.

## 8.4 Appendix D - Systems Demonstration



**Source:** http://youtu.be/jkOSz5S8neo



**Source:** http://youtu.be/zMkEP40tUc8