Areesh Nadeem | NetID: anade2 | anade2@uic.edu
Adrian Silva | NetID: asilv5 | asilv5@uic.edu

Project: WATER LEAK DETECTOR

ABSTRACT:

Our project is a real-time water leak detection system using two Arduino boards to identify and alert users to leaks, helping prevent damage. One Arduino monitors for water, while the other activates alarms through LED indicators, a buzzer, and an LCD display to notify users of leaks immediately. The user can stop the alarm by pressing a button. The Arduinos will send signals using serial communication. Unlike conventional projects that measure water levels, our focus is on rapid leak detection and immediate notification to help prevent water-related damage.

DETAILED PROJECT IDEAS:

- **Overall Description of Project Idea**

Water leaks can cause significant damage if they go unnoticed, particularly in areas like basements, kitchens, and bathrooms. Our project addresses this issue by developing a water leak detection system that utilizes two Arduino boards to detect water leaks and promptly alert users. The project aims to provide visual and audio signals in real time, so the user can quickly address the water leak. By placing this detection system in strategic areas within a home, it can minimize water damage and prevent costly repairs.

The system uses a water sensor to detect the presence of water and immediately triggers alerts. When water is detected, the system will activate LED indicators for a visual alert, and a buzzer for an auditory signal. Until a button is pressed, the alarms will keep running. Additionally, an LCD display will be included to indicate the leak status.

- **Final Project Design stating how Multiple Arduinos will be used**

This project will utilize two Arduinos, each assigned specific roles. The first Arduino will act as the detection unit. It will monitor a water sensor placed in a location where leaks are likely to occur, such as the floor of a basement or under a kitchen sink. When the sensor detects water, it will send a signal to the second Arduino. The second Arduino will be dedicated to handling the alert system. Upon receiving a signal from the first Arduino, it will activate a buzzer and flash an LED to notify the user immediately. The two Arduinos have specialized roles, with one focused only on water detection and the other managing the alerts.

- **Final Plan for Use and Communication between the multiple Arduinos**

  For this project, the two Arduinos will communicate using serial communication. The detection Arduino will send a signal to the alerting Arduino whenever water is detected. This communication method ensures that the alerting system is only triggered when necessary and remains silent otherwise. Using serial communication will also allow the two Arduinos to remain synchronized and responsive, ensuring that any leak is detected and reported quickly.

  To ensure that the user notices the alerts and addresses the leak, the detection Arduino will send repeated signals for as long as water is detected, so the alert Arduino can continuously flash the LED and activate the buzzer until the button is pressed. By doing this, we can ensure that the alert system keeps running until the leak is addressed.

- **Final Project Design stating Expected Inputs/Outputs**

  The project's input devices will consist of a water sensor, which is the primary tool for leak detection. This sensor will be calibrated to detect even small amounts of water, as the goal is to alert the user at the first sign of a leak, minimizing potential water damage. Once water is detected, a serial signal will be sent from the detection Arduino to the alert Arduino. Additionally, we will have a button to stop the alarm.

  The output devices for this project will include 2 flashing LED and a buzzer to serve as visual and auditory alerts. The LEDs will continuously flash, and the buzzer will emit a loud sound as long as the water sensor is detecting water, and it is receiving signals from the detecting Arduino. Additionally, an LCD display will be connected to the alert Arduino to provide the status of the leak (No Leak / LEAK DETECTED) as well as (BUTTON TO STOP) if the alarm is active. The combination of all these visual and audio outputs will ensure that the user is aware of the leak even if they are not near the system.

- **Final description of the original work being attempted by your project.**

  Our project focuses on real-time water leak detection and alerting, rather than just water level measurement. Many existing water sensor projects are primarily concerned with measuring the water levels in containers, soil moisture, or general water depth in aquariums, which has a different application to what we are trying to build. Our sensor will not be used to measure water levels, rather it actually won't care about the level at all. If it detects even the smallest amount, we will program it to trigger the alerts. The immediate activation of the alert system upon detection sets our project apart from other

systems that only display water levels. By modifying the water sensor use in this way, we are creating a tool that people can use for home safety.

- **How the project will be used.**

  Place the water sensor beneath areas where you want to monitor for leaks. Alternatively, place it in a cup (or any open container) to monitor a larger area for water presence. Try to keep the Arduino's out of range of any potential water by encasing them or Having a long enough connection from the water sensor to the Arduinos to move them to a safer place. Once a leak has been detected by the water sensor, The second Arduino will activate the LED, buzzer, and change the text on the LCD. Pressing the button found on the second Arduino's breadboard will deactivate the buzzer and the LED. Before placing the sensor to detect leaks again, make sure to dry it using a lint-free cloth or letting it air dry. To reset the leak detector system, simply unplug and replug the power from the alert arduino.

REQUIRED SUPPORTING MATERIALS:

- **Week by Week Timeline:**
  - **Week 11/3 - 11/9** : Get all materials needed, including materials we have to purchase (water sensor).
  - **Week 11/10 - 11/16** : Experiment with water sensor and see how it works, start designing on TinkerCAD.
  - **Week 11/17 - 11/23** : Finish up TinkerCAD Design and prepare for design presentation. Start working on assembling hardware and coding referencing the TinkerCAD design. Debug the water sensor for humidity / environment noise.
    - **11/22** : Design presentation
  - **Week 11/24 - 11/30** : Finish up hardware and code, start preparing for project demonstration.
  - **Week 12/1 - 12/6** : Finalize and touch up any needed parts on project, and practice for project demonstration. Finalize how we will be demoing the project.
    - **12/6** : Project Demonstration

- **Final List of Materials expected to be needed**

    - 2 Arduinos
    - (DIYables) Water Sensor (or brands with similar sensor)
    - 2 Red LEDs
    - LCD Display
    - Button
    - 10k ohm Potentiometer
    - Buzzer
    - Breadboard
    - Several wires
    - 3x 220 ohm resistors

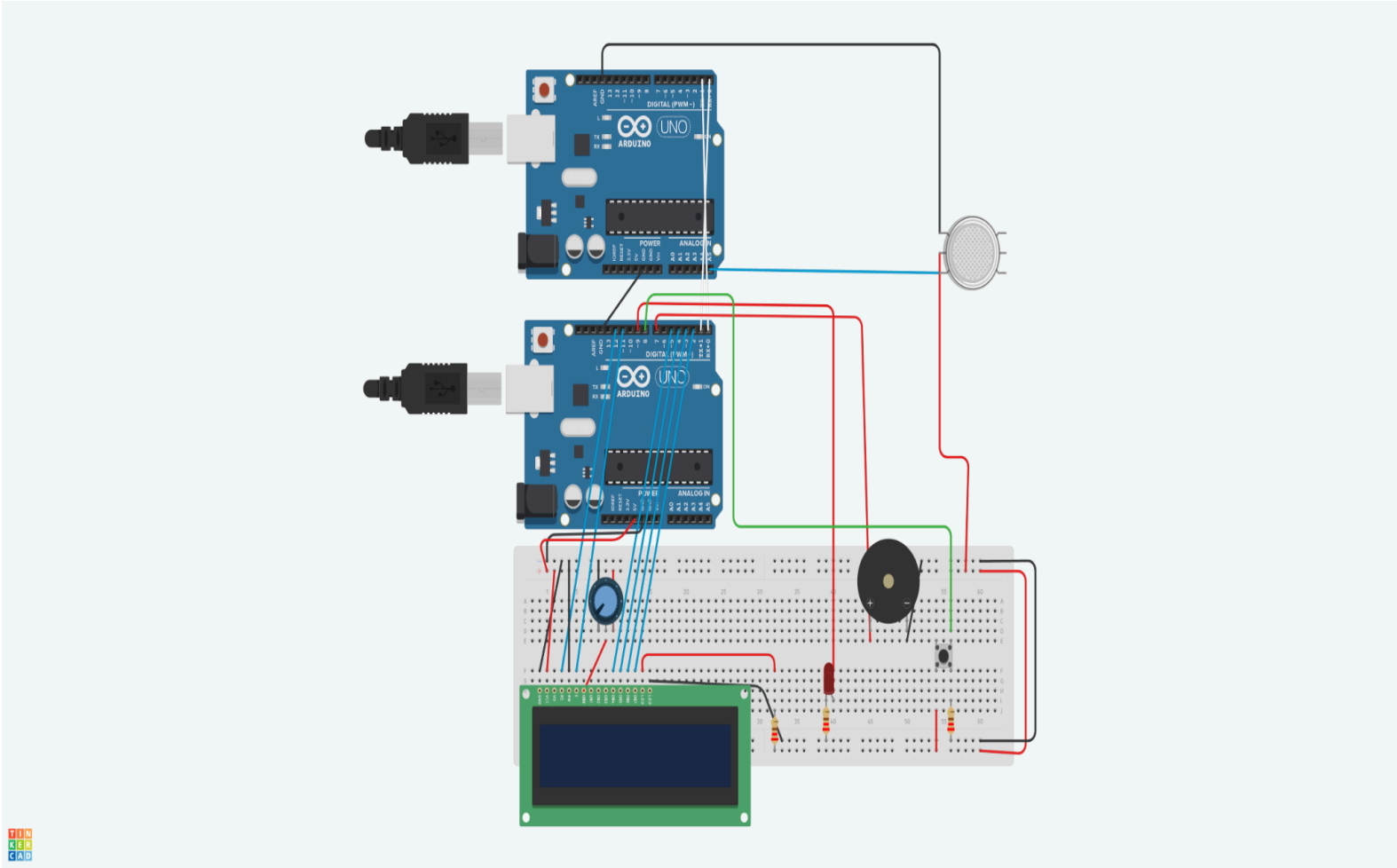
- **How to build our project (Wiring and Assembly)**

1. Arduino Communication Connections
    - Top Arduino TX (Pin 1): Connect to Bottom Arduino RX (Pin 0)
    - Top Arduino RX (Pin 0): Connect to Bottom Arduino TX (Pin 1)
    - Top Arduino Analog GND: Connect to Bottom Arduino Digital GND
2. 16x2 LCD Display Connections to Alert Arduino
    - Pin 1 (VSS): Connect to GND
    - Pin 2 (VDD): Connect to 5V
    - Pin 3 (V0): Connect to the middle pin of the potentiometer
    - Pin 4 (RS): Connect to digital pin 12
    - Pin 5 (RW): Connect to GND
    - Pin 6 (E): Connect to digital pin 11
    - Pin 11 (D4): Connect to digital pin 5
    - Pin 12 (D5): Connect to digital pin 4
    - Pin 13 (D6): Connect to digital pin 3
    - Pin 14 (D7): Connect to digital pin 2
    - Pin 15 (A): Connect to 5V through a 220Ω resistor
    - Pin 16 (K): Connect to GND
3. Potentiometer Connections
    - Left Pin: Connect to 5V
    - Middle Pin: Connect to LCD Pin 3 (V0)
    - Right Pin: Connect to GND
4. Button Connections to Alert Arduino
    - Top-right side of the button: Connect to digital pin 8
    - Left side of the button: Connect to 5V

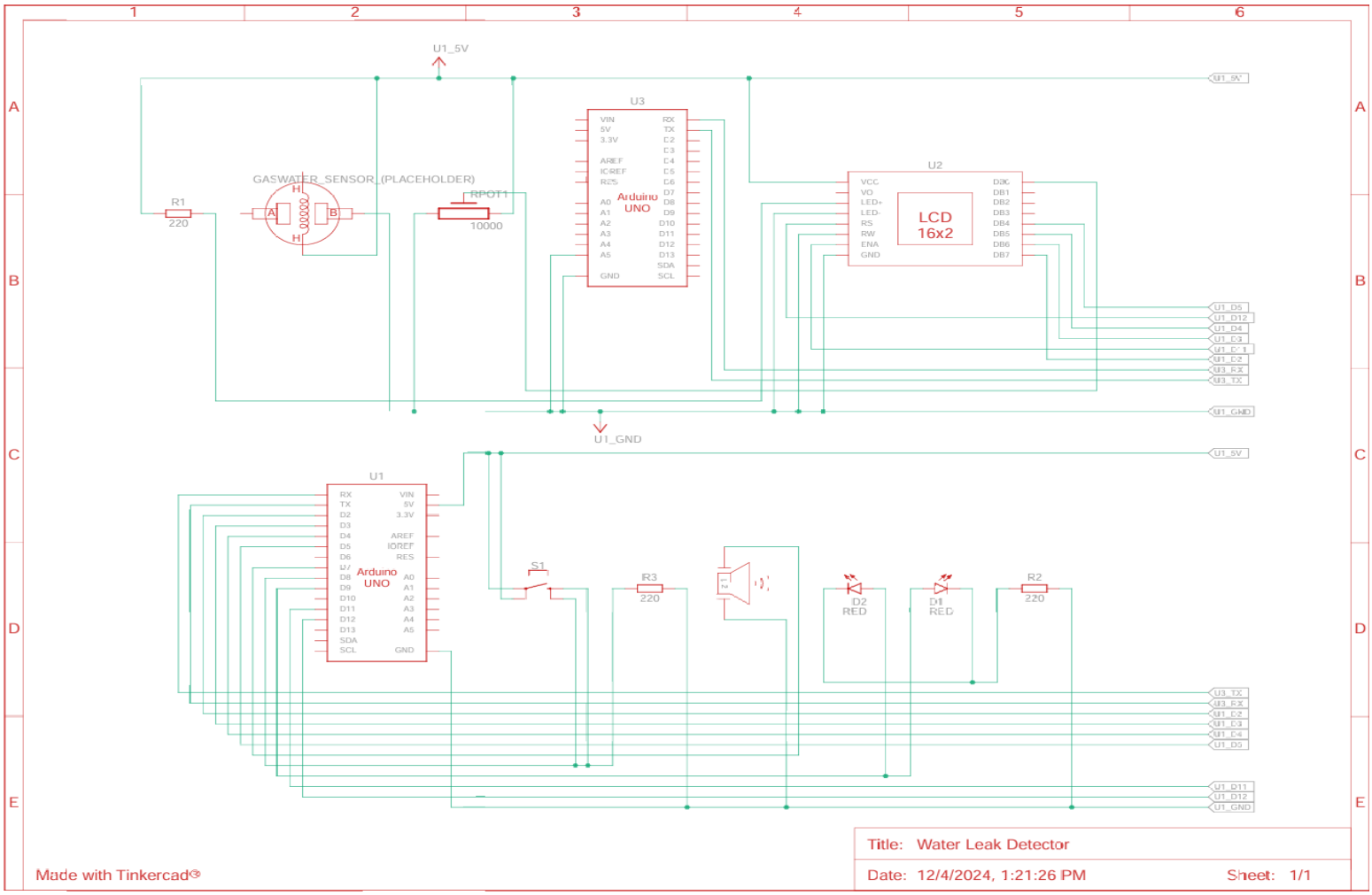
    - Bottom-right connect to GND through 220 ohm resistor
5. LED Connections to Alert Arduino
    - Place two LEDs vertically aligned to breadboard
    - Connect anode to digital pin 9
    - Connect to ground through a 220 ohm resistor
6. Buzzer Connection to Alert Arduino
    - Connect anode side to digital pin 7
    - Connect cathode to GND
7. Water Sensor Connection to Sensor Arduino
    - Connect the GND pin (top pin) to GND on Arduino
    - Connect VCC (middle pin) to 5V on breadboard
    - Connect Signal pin (bottom pin) to analog pin A5
8. Power
    - Finally, connect the 5V and GND pins on the Alert Arduino to the corresponding rails on the breadboard to supply power
    - Additionally, connect both the 5V rails and GND rails to each other on the breadboard

- **Final List of references**
    - Water Sensor
    - LCD Display
    - Buzzer
    - Button
    - Serial Communication

- **Inclusion of one or more diagrams of hardware**
  - *The **white sensor** is a placeholder for the **Water Sensor***
  - *Top pin is GND, Middle is VCC, Bottom is Sensor*



Schematic:



- **Final Code Implementation**

SENSOR ARDUINO CODE

```
// Areesh Nadeem, netID: anade2
// Adrian Silva, netID: asilv5
// Project Name: Water Leak Detector
// Sensor Arduino
/*
Abstract: Our project is a real-time water leak detection system
using two Arduino  boards to identify and alert users to leaks,
```

```
helping prevent damage. One Arduino monitors for water,  while
the other activates alarms through LED indicators, a buzzer, and
an LCD display to notify  users of leaks immediately. The user can
stop the alarm by pressing a button. The Arduinos will send signals
using serial communication. Unlike conventional projects that measure
water levels,  our focus is on rapid leak detection and immediate notification
to help prevent water-related damage.
*/

#define SIGNAL_PIN A5 // water sensor analog pin
int sensorVal = 0;

unsigned long prevMillis = 0;
const int threshold = 200;

void setup() {

 // serial communication with Alert Arduino
 Serial.begin(9600);
}

void loop() {

   if (millis() - prevMillis >= 300) {
       prevMillis = millis();

       // Read the value of the analog sensor
       sensorVal = analogRead(SIGNAL_PIN);

       // Serial.println(sensorVal); // for debugging

       // check if the sensor value is above the threshold
       if (sensorVal >= threshold) {
           // LEAK detected, send signal to Alert Arduino
           Serial.write('L');
       }

       else {
           // Send a signal indicating no leak detected
```

```
            Serial.write('N');
        }

    }
}
```

ALERT ARDUINO CODE

```
// Areesh Nadeem, netID: anade2
// Adrian Silva, netID: asilv5
// Project Name: Water Leak Detector
// Alert Arduino
/*
Abstract: Our project is a real-time water leak detection system
using two Arduino  boards to identify and alert users to leaks,
helping prevent damage. One Arduino monitors for water,  while
the other activates alarms through LED indicators, a buzzer, and
an LCD display to notify  users of leaks immediately. The user can
stop the alarm by pressing a button. The Arduinos will send signals
using serial communication. Unlike conventional projects that measure
water levels,  our focus is on rapid leak detection and immediate notification
to help prevent water-related damage.
*/


#include <LiquidCrystal.h>

// LCD Pins
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

const int ledPin = 9;
const int buzzerPin = 7;
const int buttonPin = 8;

unsigned long prevMillis = 0;
unsigned long toggleMillis = 0; // For LED and buzzer toggling
```

```
int ledState = LOW;

bool alarm = false; // if alarm is active or not
bool alarmStopped = false;

#define NOTE_C8 4186 // note for buzzer

void setup() {
   pinMode(ledPin, OUTPUT);
   pinMode(buzzerPin, OUTPUT);
   pinMode(buttonPin, INPUT);

   // Set up the LCD
   lcd.begin(16, 2);

   // serial communication with Sensor Arduino
   Serial.begin(9600);
}

void loop() {

   // check if signal is available from sensor arduino
   if (Serial.available()) {
       char signal = Serial.read();
       Serial.println(signal);

       // process the received signal
       if (signal == 'L' && !alarmStopped) {
           // If 'L' received activate the alarm
           alarm = true;
       }
       else if (signal == 'N') {
           // If 'N' received, deactivate the alarm
           digitalWrite(ledPin, LOW);
           noTone(buzzerPin);
           alarm = false;
       }
   }
```

```
  // Toggle LED and buzzer every 1 second if the alarm is active
  if (alarm && (millis() - toggleMillis >= 1000)) {
      toggleMillis = millis(); // Reset toggle timer

      ledState = !ledState;
      digitalWrite(ledPin, ledState); // Update LED

      // match timing with LEDs, if LEDs on then buzzer on
      if (ledState) { tone(buzzerPin, NOTE_C8); }
      else { noTone(buzzerPin); }
  }

  // update LCD every 300 ms
  if (millis() - prevMillis > 300) {
      prevMillis = millis();

      lcd.clear();

      if (alarm) { // if alarm is active
          lcd.setCursor(0, 0);
          lcd.print("LEAK DETECTED");
          lcd.setCursor(0, 1);
          lcd.print("BUTTON TO STOP");
      }

      else if (alarmStopped) { // alarm has been stopped
          lcd.setCursor(0, 0);
          lcd.print("Alarm Stopped");
      }

      else { // no leak
          lcd.setCursor(0, 0);
          lcd.print("No Leak");
      }
  }

  // check if alarm is active and button is pressed
  // means user stopped the alarm
  if (alarm == true && digitalRead(buttonPin) == HIGH) {
```

```
        // turn off alerts
        digitalWrite(ledPin, LOW);
        noTone(buzzerPin);

        alarm = false;
        alarmStopped = true;

        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Alarm Stopped");
    }
}
```