Security System Music Player

Milestone #4

Group members:

+ Son Nguyen - snguye49 - snguye49@uic.edu

+ Quan Tran - qtranm2 - qtranm2@uic.edu

+ Akshant Jain - ajain78 - ajain78@uic.edu

- **Abstract of the project:**

Today's modern anti-theft systems have evolved from much simpler systems, starting with the basic lock and key concept. Since the invention of this and other less complex safety mechanisms, a variety of high-tech security devices have been produced, including RFID tags and biometric identification. However, users do not currently have the option to customize the security device they want, instead, they have to choose from the devices currently available on the market. For Milestone 3, we want to introduce a customizable anti-theft product that will play the user's choice of music when the device detects a possible threat around the user's home.

- **Detailed project ideas:**

1. Overall Description of Project Idea:

The main idea for our Security System Music Player project is to create a responsive auditory security system to alert users to any credible safety threats within the range of user's home. Regarding the mechanical aspects of this system, there will be several different components responsible for the various functions being performed. The central function of this system is to play the music or melody that the user has pre-selected in response to a possible safety threat so that they are aware of the situation. In addition to the musical component of this system, we also plan on implementing several accompanying forms of visual alerts, including LEDs and LCD screens. We will also include an activation and deactivation component to our system, which is managed by using a passcode input from keypad, so that the user can enable and disable the alert system accordingly. Since the central mechanics of the system will function

by playing music based on user's preference after the alarm is triggered by an outside threat, the user will first need to set up the system by selecting a list of pre-installed melodies to be played when the system detects a threat. In addition to the motion sensor, in Milestone 4, we want to add one more feature to our system. We will include a gas/smoke detector sensor to our system and the process of delivering alert will be the same as the motion sensor. The reason is that one of our main goals for this project is maximizing the user's safety and protecting them from a variety of possible dangers. With this newly added feature, the system will not only detect thief, unwanted strangers but also detect fire, thus fulfilling the primary purpose of "security".

2 and 3. UPDATED Project Design stating how Multiple Arduinos will be used and UPDATED Plan for Use and Communication between the multiple Arduinos:

For Milestone 4, we are still planning on using three Arduino microcontrollers to enable the functions necessary for our project. Previously in Milestone 3, we were planning for those three Arduinos to be connected to different appliance and devices to cover every threat using Bluetooth to send signals to each other. However, when we try to implement this method, we discover several potential problems working with the wireless range approach, in this case, Bluetooth. The first difficulty is the ranging of the Bluetooth connection. Generally, most devices running on Bluetooth have a short-range connection as the maximum range for connecting to two devices using Bluetooth is approximately 30 feet. To have a long-distance connection, they need an external power supply such as an amplifier board or AC/DC power adapter, but these extensions are mainly used for industrial purposes and are not that cheap. In our project, since we need to locate one Arduino outside the user's house to send the signal and one Arduino inside the user's house to receive signal, the range between those two Arduinos may be more than 30 feet and they may not be able to communicate properly.

The second problem is the slow data transfer. As we all know, Bluetooth transfers data much slower than other connections like Wi-fi or serial connection. Bluetooth 4.0 can only get up to transferring 25 megabits per second max and there may be some delay when sending signals between two Arduinos with each other. This will become a bigger issue in the future as we want to alert the user immediately when the system detects theft or fire. For that reason, we decide to set up a new connection between Arduinos. To allow the Arduinos to communicate with one another and carry out their functions, we will be utilizing a serializable connection system. This system will feature a wired connection that enables the three Arduino

microcontrollers to share information and relay signals back and forth throughout the system. The reason why we change our type of connection between those Arduinos is because this type of connection will be a reliable way to ensure that desired actions of our system are carried out efficiently and will be able to transfer data up to 250 megabits per second, thus faster response time to alert the users of possible threads inside and outside their houses.

As previously mentioned in Milestone 3, communication between the Arduinos included in this project design will be facilitated by three distinct connections. The first Arduino will contribute to the system functions through a wired connection to a motion sensor. We will use a power supply for the motion sensor and have it always on, constantly listening for input. Once motion is detected, the information will be transmitted to the second Arduino that is connected to a buzzer. The buzzer will start alerting the house by buzzing for 30 seconds. After that, the motion sensor will send a signal to the third Arduino that is connected to a speaker or buzzer. The speaker will then start playing audio according to the user's choice repeatedly until the user taps the pause button to turn it off and resets it to the original state so that it can be used next time.

For Milestone 4, we designed a better way to make use of the three Arduinos. The first Arduino will be the same, but for the second Arduino, we want to attach both a buzzer and a speaker. For the third Arduino, we will connect it to an LED, an LCD, and the keypad. The user guide for this will be attached below in the User Guide section.

4. UPDATED Project Design stating Expected Inputs/Outputs:

- Expected Inputs:
    + Button should turn off the LED and stop the buzzer.
    + 4x4 keypad should input correct digits to set up and input the password.
- Expected Outputs:
    + LED should be turned on when there is a potential threat.
    + Buzzer should be turned on when there is a potential threat.
    + Photoresistor and motion sensor should output value based on their functionalities.
    + Gas sensor should return a high value when there is a smoke.

5. UPDATED Description of the original work being attempted by your project:

Nowadays, most fire alarms and theft alert systems are designed separately and are restricted in the number of available sound options. For our project, we merge two functions of detecting theft and fire into a single device package, thus making it more convenient for the user. We also aim for our project to work with no less than 10 different customizable sounds so far more complicated sounds will be produced. Additionally, we will apply everything that we learn in CS 362 labs into this project from the very basic such as LED blinking, LCD screen displaying to more advanced topics such as the serializable connection between two Arduinos and interrupt between multiple Arduinos.

6. Discussion on how to build our project (think Lab Reports)

1. For the first Arduino:

    Step 1: Pin 0 and 1 are used for serial communication between other Arduinos.

    Step 2: Pin 2 to 9 are used for receiving input from keypad.

    Step 3: Pin 10 and 11 are for receiving input from the distance sensor.

    Step 4: Pin 12 is for triggering the buzzer.

    Step 5: Pin 13 is to turn on the LED.

    Step 6: Pin A0 is for receiving value from the photoresistor.

2. For the second Arduino:

    Step 1: Pin 0 and 1 are used for serial communication between other Arduinos.

    Step 2: Pin 4 is for receiving input from a button.

    Step 3: Pin 5 is for triggering the buzzer.

    Step 4: Pin 6 is to turn on the LED.

    Step 5: Pin 7 to 12 is for connecting with the LCD.

3. For the third Arduino:

    Step 1: Pin 0 and 1 are used for serial communication between other Arduinos.

    Step 2: Pin 4 is for receiving input from a button.

    Step 3: Pin 5 is for triggering the buzzer.

    Step 4: Pin 6 is to turn on the LED.

    Step 5: Pin 7 to 12 is for connecting with the LCD.

7. Discussion on how our project is to be used (think User Guide)

1. The user sets up a passcode and puts the first Arduino at the front door of the house.
2. The user put the second Arduino and the third Arduino anywhere inside the house. Those 2 Arduinos will need to be set up close together.
3. The user then chooses sound preference to play from the buzzer for all 3 Arduinos and the setup is done
4. The photoresistor from the first Arduino should be activated at night.
5. When the motion sensor from the first Arduino detects someone around, it will send a signal to the second Arduino and the third Arduino
6. All three Arduinos will then start blinking LED to indicate that there is a ongoing potential threat around the house
7. The LCD screen on the second and third Arduino will also start displaying messages on them, indication general information like how long the buzzer have been buzzing, etc...
8. The same concept will also work if the gas sensor from the third Arduino senses a potential fire around the house
9. To stop the buzzing sound, the user will need to click the button from the second and third Arduino, which is located inside user's house.

- **REQUIRED Supporting Materials:**

1. **Timeline of Development given in a week-by-week format of completed items**

+ Week 1: Form groups and start sharing each other idea.

+ Week 2: Learn how to wire a LED.

+ Week 3: Learn how to use buttons.

+ Week 4: Learn how to use an LCD display.

+ Week 5: Learn how to use a photoresistor.

+ Week 6: Learn how to utilize the serial communication.

+ Week 7: Learn how to apply serial communication for two Arduino boards

+ Week 8: Learn about interrupts between two Arduinos.

+ Week 9: Start finalizing out project ideas.

+ Week 10: Watch tutorials online and start the wiring and basic coding.

+ Week 11: Draw diagrams and finish the documentation for the project

2. **Timeline of Development given in a week-by-week format of work still to do**

+ Week 12: Start coding for the first Arduino which will include LEDs, buzzer, buttons, gas sensor and an LCD screen.

+ Week 13: Start coding for the second Arduino which will include LED, buttons, buzzer, and an LCD screen.

+ Week 14: Start coding for the third Arduino which will include a motion sensor, buzzer, keypad and a photoresistor.

+ Week 15: Try to communicate and connect all the three Arduinos together.

+ Week 16: Finish everything left that needs to be done and demo our project.

3. **UPDATED List of Materials Expected to be Needed**

+ 3 Arduino microcontrollers

+ 3 LEDs

+ 2 buttons

+ 1 gas sensor for detecting smoke hazard

+ 1 keypad

+ 3 buzzers

+ 1 photoresistor

+ 1 motion sensor for detecting motion/distance

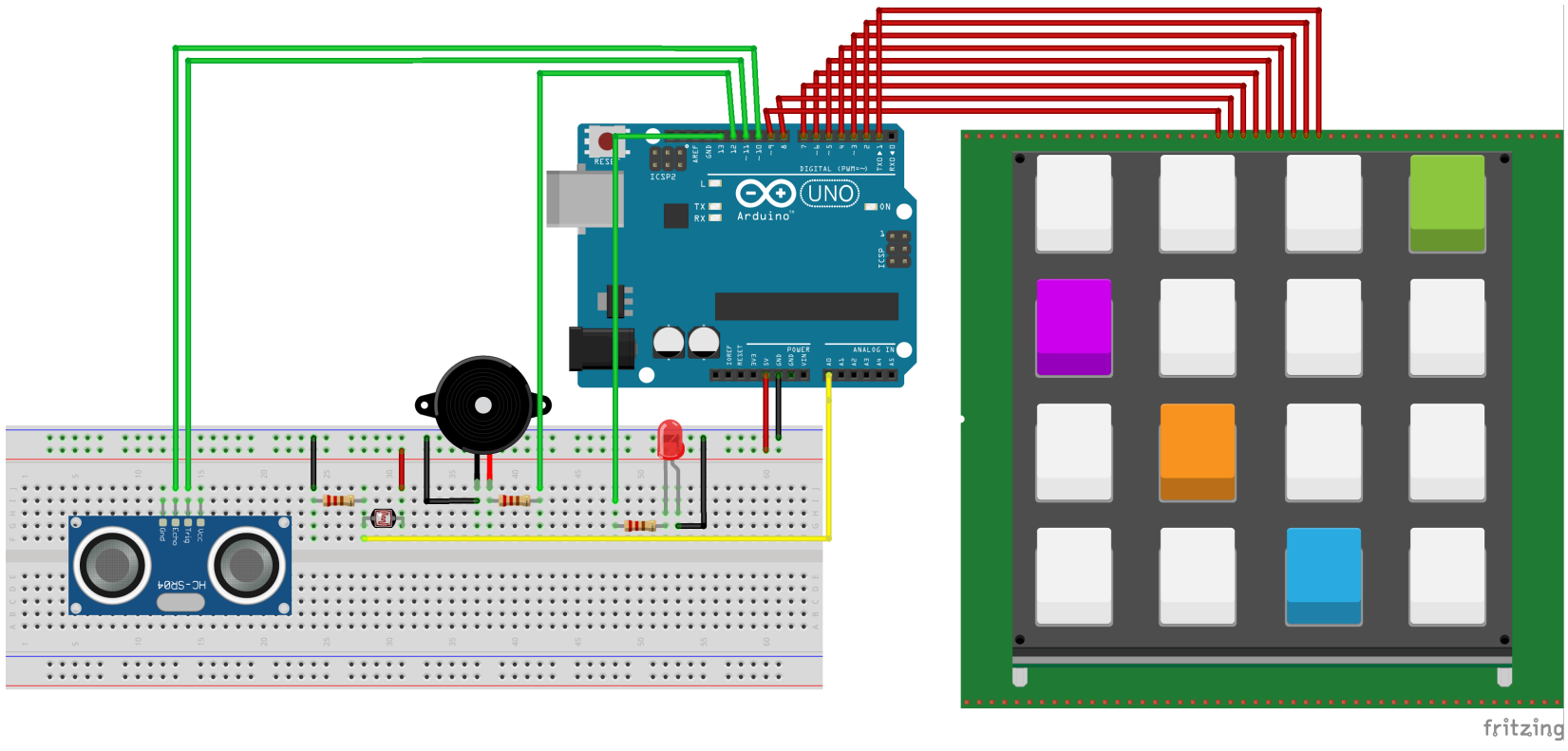
+ 2 LCD screens
+ 2 Potentiometers
+ Breadboards
+ Wires
+ Battery
+ Power sources
+ Custom knobs

4. UPDATED List of References

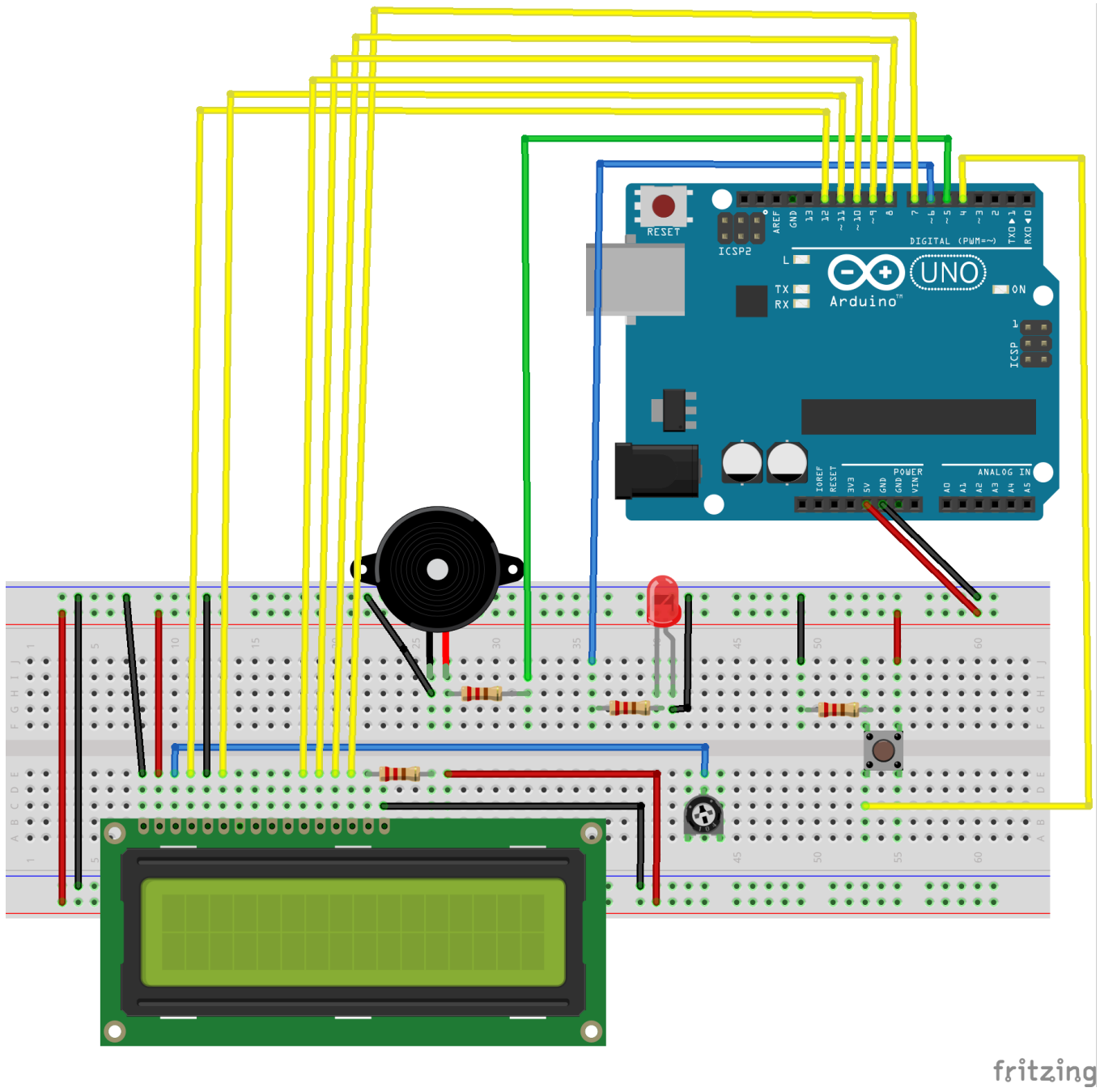
- http://arduino.cc/en/Tutorial/Blink
- https://www.arduino.cc/en/Tutorial/Button
- https://www.arduino.cc/en/Tutorial/LiquidCrystalScroll
- https://playground.arduino.cc/Learning/PhotoResistor/
- https://www.arduino.cc/en/Reference/AnalogWrite
- https://www.arduino.cc/reference/en/language/functions/communication/serial/
- https://iotguider.in/arduino/serial-communication-between-two-arduino-boards/
- http://arduino.cc/en/Reference/attachInterrupt
- https://learn.sparkfun.com/tutorials/connecting-arduino-to-processing#introduction
- https://circuitdigest.com/microcontroller-projects/arduino-audio-music- player)
- https://create.arduino.cc/projecthub/fradirosa00/arduino-fire-alarm-4da798
- https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/
- https://create.arduino.cc/projecthub/techmirtz/using-4x4-keypad-with-arduino-2d22e9
- http://electronicsprojectshub.com/how-to-connect-mq2-gas-sensor-to-arduino/

5. UPDATED Diagrams

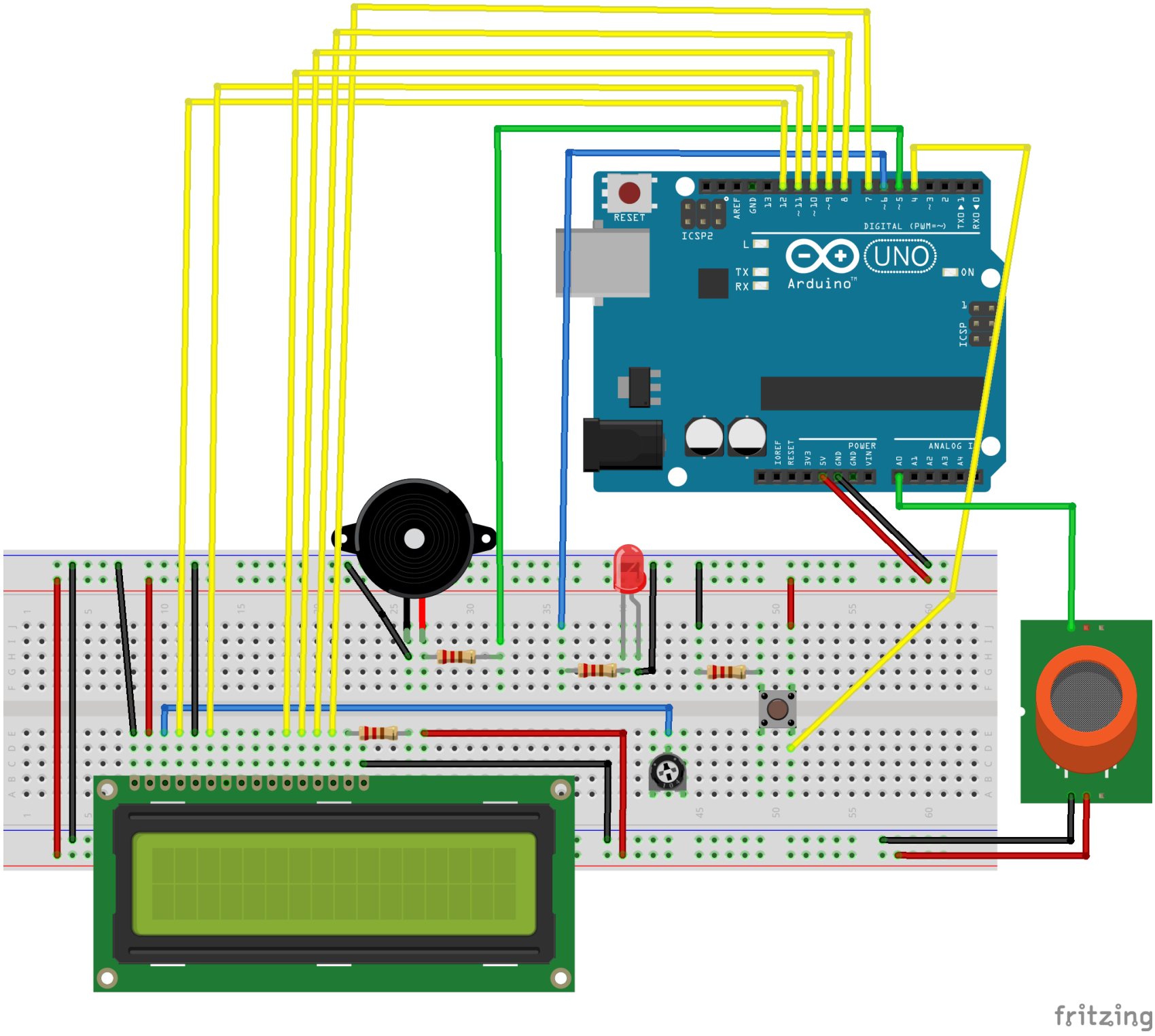
- Diagram for the first Arduino



- Diagram for the second Arduino



- Diagram for the third Arduino



6. **UPDATED** Code

- Sketch for the first Arduino

```
/* ------------------ FIRST ARDUINO ------------------ */
typedef unsigned int uint;

#include <Keypad.h>
#include <LiquidCrystal.h>
#include <Wire.h>

// First Arduino should contain
// - LED to notify user
// - Keypad to input password for activate/deactivate the system
// - Motion sensor/Distance sensor, put near the window/front door to detect a person
// - Photoresistor, put near the window/front door to detect brightness in midnight
// - Buzzer to play a sound when system detects a potential threat

static const uint LED_PIN = 13;     // led pin
static const uint BUZZER_PIN = 12;  // piezo buzzer pin
static const uint PHOTO_PIN = A0;   // photoresistor pin

// for ultrasonic sensor
static const uint TRIG_PIN = 11;
static const uint ECHO_PIN = 10;

/* ---- Keypad ---- */
const char keymap[4][4] = {
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {'*', '0', '#', 'D'}};

byte rowKeyPins[4] = {'9', '8', '7', '6'};
byte colKeyPins[4] = {'5', '4', '3', '2'};

Keypad keypad = Keypad(makeKeymap(keymap), rowKeyPins, colKeyPins, 4, 4);

// piezo buzzer frequency response
// play midi to piezo buzzer

/* ------------------------------------------------------ */

void triggerBuzzer() {
```

```
  // TRIGGER THE BUZZER
  // notes to play a warning sound
  const int[] notes = {440, 880, 1760};

  // change the tone of the buzzer based on the potentiometer
  tone(BUZZER_PIN, notes[0]);
  tone(BUZZER_PIN, notes[1]);
  tone(BUZZER_PIN, notes[2]);
}

void setup() {
  pinMode(LED_PIN, OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(PHOTO_PIN, INPUT); // analog?
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  Serial.begin(9600);
}

void loop() {
  /*** HANDLE INPUT FROM KEYPAD ***/
  // Initial state: Ask user to setup a password, confirm twice then system is activated
  // State 1: Waiting for input from user
  //          IF USER ENTERS PASSWORD INCORRECT 5 TIMES, TRIGGER THE BUZZER
  // State 2: Buzzer is deactivated by enter the correct password, go back to State 1,
else keep buzzing.
  char keyPressed = keypad.getKey();
  if (keyPressed != NO_KEY) {
    Serial.println(keyPressed);  // what key was pressed?
  }
  // ...

  /*** HANDLE MOTION/DISTANCE SENSOR ***/
  // clear the TRIG_PIN
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  // set the TRIG_PIN on HIGH state for 10ms
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  // read the ECHO_PIN, returns the sound wave travel time in microseconds
  duration = pulseIn(ECHO_PIN, HIGH);
  // calculating the distance
```

```
  distance = duration * 0.034 / 2;
  // print the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.println(distance);

  // set threshold for distance, if distance get too low for too long, trigger the buzzer
  if (distance < 10) {
    triggerBuzzer();
  }

  /*** HANDLE PHOTORESISTOR ***/
  // set threshold for photoresistor, if it gets too bright, trigger the buzzer
  if (analogRead(PHOTO_PIN) > 800) {
    triggerBuzzer();
  }
}
```

- Sketch for the second Arduino

```
/* ------------------ SECOND ARDUINO ------------------ */
typedef unsigned int uint;

#include <LiquidCrystal.h>
#include <Wire.h>

// Second Arduino should contain
// - LCD to notify user
// - LED to notify user
// - Buzzer to play a sound when system detects a potential threat

static LiquidCrystal lcd(12, 11, 10, 9, 8, 7);  // lcd 16x2

static const uint LED_PIN = 6;     // led pin
static const uint BUZZER_PIN = 5;  // buzzer pin
static const uint BUTTON_PIN = 4;  // button pin

/* ---------------------------------------------------- */

void triggerBuzzer() {
  // TRIGGER THE BUZZER
  // notes to play a warning sound
  const int[] notes = {440, 880, 1760};

  // change the tone of the buzzer based on the potentiometer
  tone(BUZZER_PIN, notes[0]);
  tone(BUZZER_PIN, notes[1]);
  tone(BUZZER_PIN, notes[2]);
}

void setup() {
  lcd.begin(16, 2);

  pinMode(LED_PIN, OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(BUTTON_PIN, INPUT);
  Serial.begin(9600);
}

void loop() {
  // receive information from the first Arduino and acts accordingly
```

```
  int receivedData = Serial.read();
  // output receivedData to the LCD to notify to the user
  // show the password, show the value from sensors
  lcd.write(receivedData);

  // received a threat, trigger the buzzer
  triggerBuzzer();

  // if the buzzer is triggered, push a button should stop the buzzer, turn off the led
  int button = digitalRead(BUTTON_PIN);
  if (button == HIGH) {
    // stop the buzzer
    // turn off the LED
  }
}
```

- Sketch for the third Arduino

```
/* ------------------ THIRD ARDUINO ------------------ */
typedef unsigned int uint;

#include <LiquidCrystal.h>
#include <Wire.h>

// Second Arduino should contain
// - LCD to notify user
// - LED to notify user
// - Buzzer to play a sound when system detects a potential threat

static LiquidCrystal lcd(12, 11, 10, 9, 8, 7);  // lcd 16x2

static const uint LED_PIN = 6;     // led pin
static const uint BUZZER_PIN = 5;  // buzzer pin
static const uint BUTTON_PIN = 4;  // button pin
static const uint GAS_PIN = A0;    // gas pin

int lpg, co, smoke;
MQ2 mq2(GAS_PIN); // gas

// piezo buzzer frequency response and play midi to piezo buzzer

/* ------------------------------------------------------ */

void triggerBuzzer() {
  // TRIGGER THE BUZZER
  // notes to play a warning sound
  const int[] notes = {440, 880, 1760};

  // change the tone of the buzzer based on the potentiometer
  tone(BUZZER_PIN, notes[0]);
  tone(BUZZER_PIN, notes[1]);
  tone(BUZZER_PIN, notes[2]);
}

void setup() {
  lcd.begin(16, 2);
  mq2.begin();

  pinMode(LED_PIN, OUTPUT);
```

```
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(BUTTON_PIN, INPUT);
  Serial.begin(9600);
}

void loop() {
  // receive information from the first Arduino and acts accordingly
  int receivedData = Serial.read();
  // output receivedData to the LCD to notify to the user
  // show the password, show the value from sensors
  lcd.write(receivedData);

  // received a threat, trigger the buzzer
  triggerBuzzer();

  // if the buzzer is triggered, push a button should stop the buzzer, turn off the led
  int button = digitalRead(BUTTON_PIN);
  if (button == HIGH) {
    // stop the buzzer
    // turn off the LED
  }

  /*** HANDLE GAS SENSOR ***/
  // set it false to not print the values in the Serial
  float* values = mq2.read(true);
  //lpg = values[0];
  lpg = mq2.readLPG();
  //co = values[1];
  co = mq2.readCO();
  //smoke = values[2];
  smoke = mq2.readSmoke();
  lcd.setCursor(0,0);
  lcd.print("LPG:");
  lcd.print(lpg);
  lcd.print(" CO:");
  lcd.print(co);
  lcd.setCursor(0,1);
  lcd.print("SMOKE:");
  lcd.print(smoke);
  lcd.print(" PPM");
  delay(1000);
}
```