Project ASL Translator

Team Members

1. Nathen Priyonggo - rpriyo2 - rpriyo2@uic.edu
2. Kyle Sabo - ksabo4 - ksabo4@uic.edu

Abstract

This project presents an innovative text communication device using ultrasonic sensors to convert hand distances into letters of the alphabet. Users compose messages by confirming letters with button presses and transmitting the completed text to a second Arduino via serial communication. The recipient Arduino displays the message on an LCD, enabling efficient and accessible communication without a traditional keyboard. Designed for individuals with limited mobility or those seeking alternative input methods, this system leverages intuitive sensor technology to offer an inclusive, user-friendly solution for text-based interaction. It combines simplicity and functionality to bridge communication gaps effectively.

Detailed Project Ideas

1. Overall Description of Project Idea

   This project provides an accessible, keyboard-free text messaging solution by leveraging ultrasonic sensors to measure hand distances and map them to letters of the alphabet. Users compose messages through intuitive hand gestures and button presses. The system features two Arduinos, each capable of independently composing, sending, receiving, and displaying messages. This setup facilitates bidirectional communication, making it an ideal tool for users seeking a nontraditional, user-friendly text input method.

2. Final Project Design stating how Multiple Arduinos will be used

   The project employs two Arduinos, each equipped with an ultrasonic sensor to detect hand distances for letter selection, an LCD display, to show the composed message in real-time, and push buttons to confirm letters and send completed messages.

   Each Arduino functions independently to compose messages. Once a user finalizes a message, it is transmitted to the other Arduino via serial communication. The receiving Arduino displays the message on its own LCD display. This design enables both Arduinos to send and receive messages, supporting bidirectional communication and ensuring seamless interaction.

3. Final Plan for Use and Communication between the Multiple Arduinos

   The project enables bidirectional communication between two Arduinos, each operating as an independent text messaging device. Both Arduinos are equipped with an ultrasonic sensor and LCD display, allowing users to compose and send messages from either device. This setup ensures a seamless exchange of text messages without reliance on a traditional keyboard.

   Each Arduino detects hand distance using its ultrasonic sensor and maps it to a corresponding letter of the alphabet. The selected letter is displayed on the LCD screen, where the user can confirm it by pressing a button. This process continues until the user completes a word or sentence, which is then transmitted to the other Arduino using serial communication.

   The receiving Arduino captures the transmitted message and displays it on its LCD screen. This ensures that both users can send and receive text messages in real-time, promoting efficient two-way communication. By combining intuitive hand gesture input with reliable serial communication, the system provides a robust and accessible solution for text-based interaction.

4. Final Project Design stating Expected Inputs/Outputs

    Inputs:

    - Distance measurements from the ultrasonic sensors.
    - Push buttons for:
        1. Button 1: Confirm and save a letter.
        2. Button 2: Send the composed message.

    Outputs:

    - LCD Display on each Arduino which:
        1. Shows the real-time composition of the message.
        2. Received message from the other Arduino.
    - LED Bulb to indicate message received status.

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

    This project introduces a unique, bidirectional text communication system. Unlike traditional devices, this system eliminates the need for keyboards, using ultrasonic sensors to measure hand distances and map them to letters. Each device functions independently while maintaining seamless communication via serial data exchange. The innovative design ensures accessibility, user-friendliness, and portability, making it ideal for diverse use cases, including users with limited mobility.

6. Discussion on how to build your project

    a. To wire the LED display to the board, connect the following pins:
        - Connect the LCD pin VSS to the GND rail on the breadboard.
        - Connect the LCD pin VDD to the 5V rail on the breadboard.
        - Connect the LCD pin V0 to the middle pin of the Potentiometer.
        - Connect the Potentiometer pins to the 5V and the GND rail on the breadboard separately.
        - Connect the LCD pin RS to Digital Pin 11 of the Arduino.
        - Connect the LCD pin RW to the GND rail on the breadboard.
        - Connect the LCD pin Enable (E) to Digital Pin 10 of the Arduino.

- Connect the LCD pin D4 to Digital Pin 9 of the Arduino.
- Connect the LCD pin D5 to Digital Pin 8 of the Arduino.
- Connect the LCD pin D6 to Digital Pin 7 of the Arduino.
- Connect the LCD pin D7 to Digital Pin 6 of the Arduino.
- Connect the LCD pin LED+ (A) to the 5V rail on the breadboard through a 220 Ohm resistor.
- Connect the 220 Ohm resistor to the GND rail on the breadboard.
- Connect the LCD pin LED- (K) to the GND rail on the breadboard.

b. To wire the ultrasonic sensor to the board, connect the following pins:
   - Connect the VCC pin of the ultrasonic sensor to the 5V rail on the breadboard.
   - Connect the GND pin of the ultrasonic sensor to the GND rail on the breadboard.
   - Connect the Trig pin of the ultrasonic sensor to Digital Pin 13 of the Arduino.
   - Connect the Echo pin of the ultrasonic sensor to Digital Pin 12 of the Arduino.

c. To wire the push buttons to the board, connect the following pins:
   - Connect one leg of Button 1 to Digital Pin 3 of the Arduino and through a 10k Ohm resistor to the GND rail of the breadboard.
   - Connect the other leg of Button 1 to the 5V rail on the breadboard.
   - Connect one leg of Button 2 to Digital Pin 2 of the Arduino and through a 10k Ohm resistor to the GND rail of the breadboard.
   - Connect the other leg of Button 2 to the 5V rail on the breadboard.

d. To wire the LED bulb to the board, connect the following pins:
   - Connect the positive leg (anode) of the LED bulb to Digital Pin 5 of the Arduino.
   - Connect the negative leg (cathode) of the LED bulb to the GND rail on the breadboard through a 220 Ohm resistor.

7. Discussion on how your project is to be used
    1. Place your hand in front of the ultrasonic sensor to select a letter based on distance.
    2. Press button 1 to confirm and save the letter.
    3. Repeat steps 1 and 2 to compose a word or a sentence.
    4. Press button 2 to send the message to the other Arduino.
    5. View the received message on the LCD display of the other Arduino.

Timeline of Development

Week 1	10/28/24	Research in Bluetooth/wireless communications
	10/30/24	Individual progress on Bluetooth/wireless communications
	11/01/24	Troubleshooting Bluetooth/wireless communications
Week 2	11/04/24	Started implementation of Speech-to-Text
	11/08/24	Troubleshooting of Speech-to-Text devices
Week 3	11/11/24	Started implementation of sign-to-text conversion with flex sensors
	11/15/24	Troubleshooting with flex sensors
Week 4	11/18/24	Finalized project idea and decided on using ultrasonic sensors instead of flex sensors
Week 5	11/25/24	Continued on sign-to-text conversion code with ultrasonic sensors
Week 6	12/02/24	Coded sign-to-text conversion logic and debugged serial communication
	12/04/24	Finalized project
	12/06/24	Final project demonstration

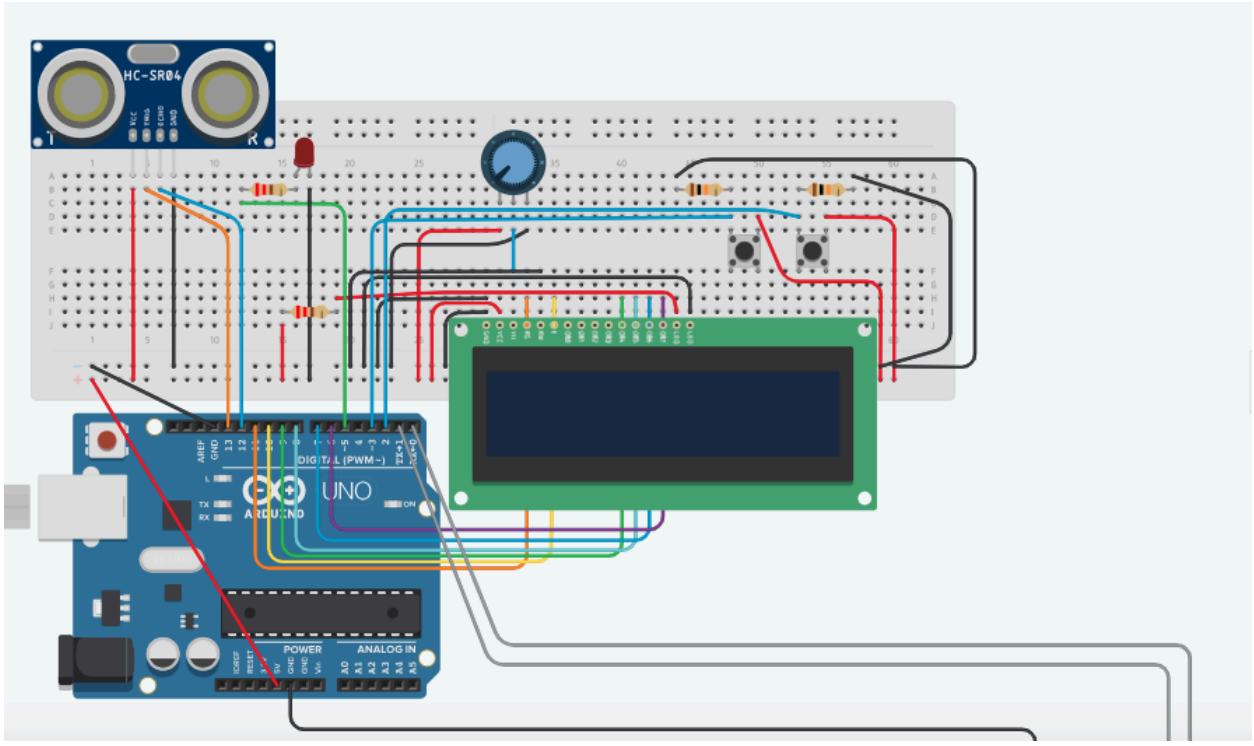
Final List of Materials

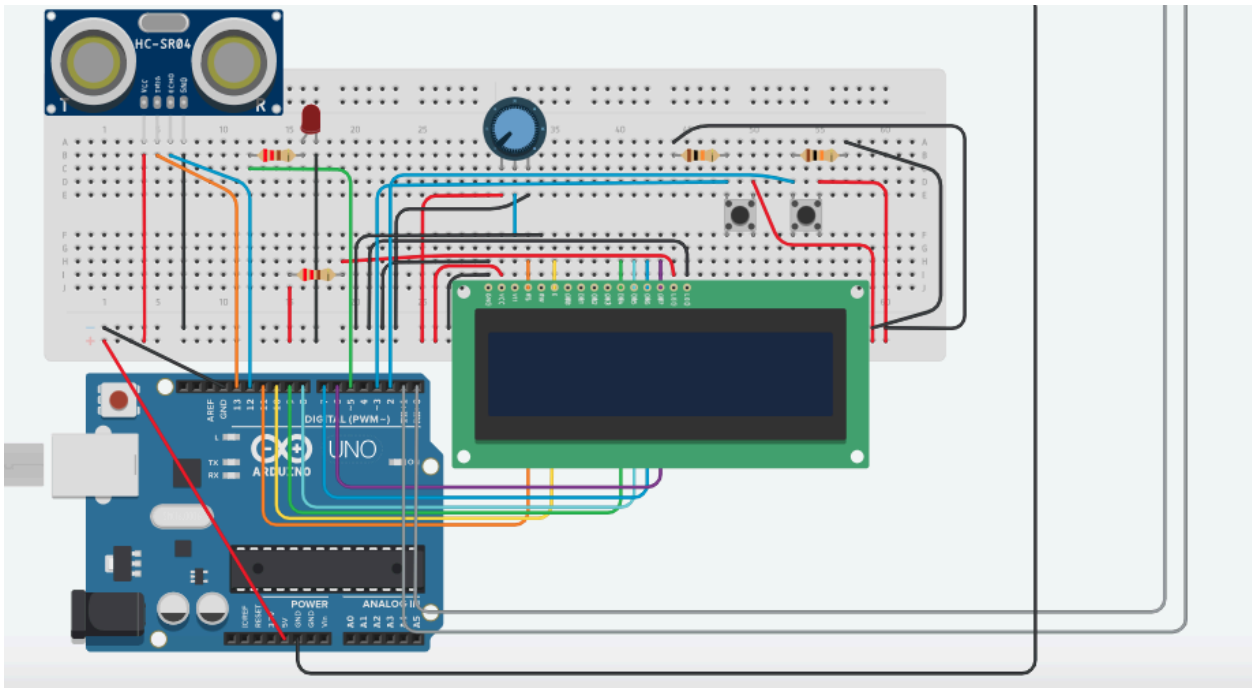
- 2x Arduinos
- 2x Breadboard
- 2x Ultrasonic sensors
- 2x LCD Displays
- 2x Potentiometer
- 4x Pushbuttons
- 2x LEDs
- 4x 10k Ohm Resistor
- 4x 220 Ohm Resistor
- Wires

Final List of References

- https://www.instructables.com/Tutorial-Using-HC06-Bluetooth-to-Serial-Wireless-U-1/
- https://www.youtube.com/watch?v=n-gJ00GTsNg
- https://docs.arduino.cc/built-in-examples/basics/Blink/
- https://docs.arduino.cc/built-in-examples/digital/Button/
- https://docs.arduino.cc/built-in-examples/digital/Debounce/
- https://docs.arduino.cc/learn/electronics/lcd-displays/

Final Diagrams





Final Code Sketches

```
/*
Team Number:
???

Team Members:
1. Nathen Priyonggo - rpriyo2
2. Kyle Sabo - ksabo4

Project Name:
Project ASL Translator

Abstract:
This project presents an innovative text communication device
using ultrasonic sensors to convert hand distances into letters
of the alphabet. Users compose messages by confirming letters
with button presses and transmitting the completed text to a
second Arduino via serial communication. The recipient Arduino
displays the message on an LCD, enabling efficient and accessible
communication without a traditional keyboard. Designed for
individuals with limited mobility or those seeking alternative
input methods, this system leverages intuitive sensor technology
to offer an inclusive, user-friendly solution for text-based
interaction. It combines simplicity and functionality to bridge
communication gaps effectively.
*/

// include library code
#include <LiquidCrystal.h>

// these constants won't change.
const int trigPin = 13;
const int echoPin = 12;
const int rs = 11;
const int en = 10;
const int d4 = 9;
const int d5 = 8;
const int d6 = 7;
const int d7 = 6;
const int ledPin = 5;
```

```
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
const int interruptPin1 = 3;
const int interruptPin2 = 2;
String sentence = "";

// these variables will change.
unsigned long prevMillis = 0;
unsigned long curMillis = 0;
unsigned long pulseStartTime = 0;
unsigned long lastInterruptTime = 0;
unsigned long lastBlinkStartTime = 0;
int pulseState = 0;  // 0 = LOW start, 1 = HIGH pulse, 2 = LOW end
int ind = 0;
int prevInd = 0;
int serialInd = 0;
bool doBlink = false;

void setup() {
 // initialize serial communication:
 Serial.begin(9600);
 pinMode(trigPin, OUTPUT);
 pinMode(echoPin, INPUT);
 pinMode(ledPin, OUTPUT);
 digitalWrite(ledPin, LOW);
  attachInterrupt(digitalPinToInterrupt(interruptPin1), ISR1, FALLING);
 attachInterrupt(digitalPinToInterrupt(interruptPin2), ISR2, FALLING);

 lcd.begin(16, 2);
 lcd.setCursor(0, 0);
}

void loop() {
 // store current time
 curMillis = millis();

 // Ultrasonic pulse logic
 if ((curMillis - prevMillis) > 50) {
   prevMillis = curMillis;

   // Trigger the ultrasonic sensor with a state-based non-blocking pulse
   if (pulseState == 0 && (curMillis - pulseStartTime) >= 2) {
     // Step 1: Set trigger pin LOW for a clean pulse start
```

```
    digitalWrite(trigPin, LOW);
    pulseStartTime = millis();
    pulseState = 1;  // Move to next state
  }
  else if (pulseState == 1 && (curMillis - pulseStartTime) >= 2) {
    // Step 2: Set trigger pin HIGH after pulse delay
    digitalWrite(trigPin, HIGH);
    pulseStartTime = millis();
    pulseState = 2;  // Move to next state
  }
  else if (pulseState == 2 && (curMillis - pulseStartTime) >= 10) {
    // Step 3: Set trigger pin LOW again after the delay
    digitalWrite(trigPin, LOW);
    pulseState = 0;  // Reset state

    // Measure the duration of the echo
    long duration = pulseIn(echoPin, HIGH);

    // Convert duration to distance
    long cm = microsecondsToCentimeters(duration);

    // Use alphaLibrary function to get character based on distance
    char alpha = alphaLibrary(cm);

    if (prevInd != ind) {
      sentence.concat(alpha);
      prevInd = ind;
    }
    // Serial.print(", ");
    lcd.setCursor(ind, 1);
    lcd.print(alpha);
  }
}

// Handle incoming  serial data
while (Serial.available()) {

  if (serialInd == 0) {
    lcd.setCursor(0, 0);
    for (int i = 0; i < 16; i++) {
      lcd.print(" ");
    }
```

```
    lcd.setCursor(serialInd, 0);
  }

  char receivedChar = Serial.read();
  // End of sentence
  if (receivedChar == '\n') {
    lcd.setCursor(ind, 1);
    serialInd = 0;
  }
  else {
    lcd.setCursor(serialInd, 0);
    lcd.print(receivedChar);
    serialInd += 1;
  }

  lastBlinkStartTime = millis();
  digitalWrite(ledPin, HIGH);
 }

 // Blink code for notification
 if (millis() - lastBlinkStartTime > 500) {
  digitalWrite(ledPin, LOW);
 }
}

long microsecondsToCentimeters(long microseconds) {
 // Formula : Distance = Time  * Speed of sound  / 2
 // The speed of sound is 340 m/s or 0.034 centimeters per microsecond.
 // The ping travels out and back, so to find the distance of the object we
 // take half of the distance traveled.
 return microseconds * 0.034 / 2;
}

char alphaLibrary(long distance) {
 char alpha;

 if (distance < 3) {alpha = 'a';}
 else if (distance < 5) {alpha = 'b';}
 else if (distance < 7) {alpha = 'c';}
 else if (distance < 9) {alpha = 'd';}
 else if (distance < 11) {alpha = 'e';}
 else if (distance < 13) {alpha = 'f';}
```

```
 else if (distance < 15) {alpha = 'g';}
 else if (distance < 17) {alpha = 'h';}
 else if (distance < 19) {alpha = 'i';}
 else if (distance < 21) {alpha = 'j';}
 else if (distance < 23) {alpha = 'k';}
 else if (distance < 25) {alpha = 'l';}
 else if (distance < 27) {alpha = 'm';}
 else if (distance < 29) {alpha = 'n';}
 else if (distance < 31) {alpha = 'o';}
 else if (distance < 33) {alpha = 'p';}
 else if (distance < 35) {alpha = 'q';}
 else if (distance < 37) {alpha = 'r';}
 else if (distance < 39) {alpha = 's';}
 else if (distance < 41) {alpha = 't';}
 else if (distance < 43) {alpha = 'u';}
 else if (distance < 45) {alpha = 'v';}
 else if (distance < 47) {alpha = 'w';}
 else if (distance < 49) {alpha = 'x';}
 else if (distance < 51) {alpha = 'y';}
 else {alpha = 'z';}

 return alpha;
}

void ISR1() {
  curMillis = millis();
 // Debounce delay for pushbutton
 if (curMillis - lastInterruptTime > 100) {
   lastInterruptTime = curMillis;
   if (ind == 15) {
     ind = 0;
     sentence = "";
   }
   else {
     ind += 1;
   }
 }
}


void ISR2() {
  curMillis = millis();
```

```
// Debounce delay for pushbutton
if (curMillis - lastInterruptTime > 100) {
  lastInterruptTime = curMillis;

  for (int i = 0; i < sentence.length(); i++) {
    Serial.write(sentence[i]);
  }
  Serial.write('\n');

  sentence = "";
  lcd.setCursor(0, 1);
  lcd.print("                ");
  lcd.setCursor(0, 1);
  ind = 0;
  prevInd = 0;
 }
}
```