Project Title: Pandemic Queue Distancing Alert using HC-SR04 Ultrasonic sensor

Team Members:

- Karl Escobar
- Ken Acula
- Angel Lopez

## 1. Background

Due to the recent outbreak of the deadly coronavirus illness in 2019, which has spread rapidly over the world, physical or social distance has become a must to avoid intimate contact. As of January 2021, there are 10.8 million impacted patients, 10.5 million recovered cases, and a death rate of 155 thousand. Businesses have reduced their in-store capacity as a result of social distancing, resulting in huge lineups of customers just attempting to get into the store. Many restaurants are still closed to indoor dining, so people are queuing for take-out orders. Customers are queuing outside of hair salons, doctors' offices, and auto repair shops, while client waiting spaces have been shuttered to reduce face-to-face contact.

The researchers developed a social distancing detection tool that can keep track of safe distances and determine the distance between people in order to reduce the impact of the coronavirus pandemic by analyzing data in real-time. This tool can be added to the security system at hospitals, workplaces, markets, jeweler's shops, and other locations to monitor social distances among people.

## 2. Project Description

This project uses HC-SR04 to measure the distance value between the object and the sensor itself. The clock will count to check if there are any changes in the distance value. The distance value that has been measured will be displayed on the LCD. A warning message will be displayed on the LCD if the value exceeds the given distance. Simultaneously, the distance detected by the sensor will be displayed, but only when a certain range has been violated will the warning message appear, not the distance value anymore.

## 3. Project Specification

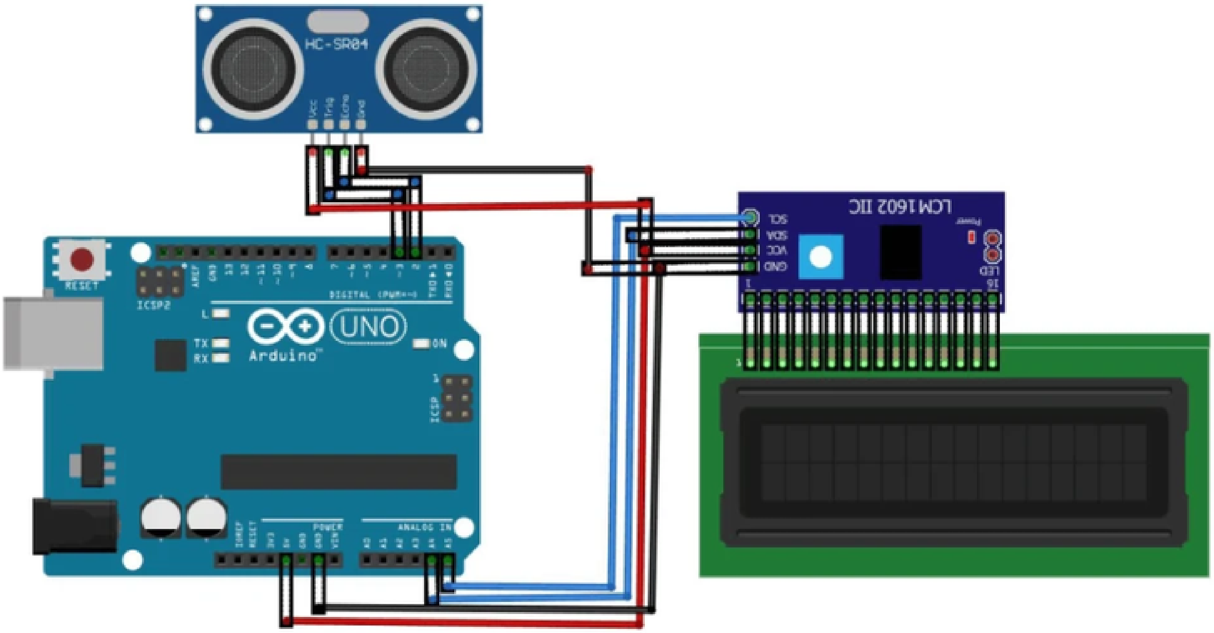
The components used for the project are:

- LCD with LCM1602 IIC
- Arduino uno
- HC-SR04 Sensor

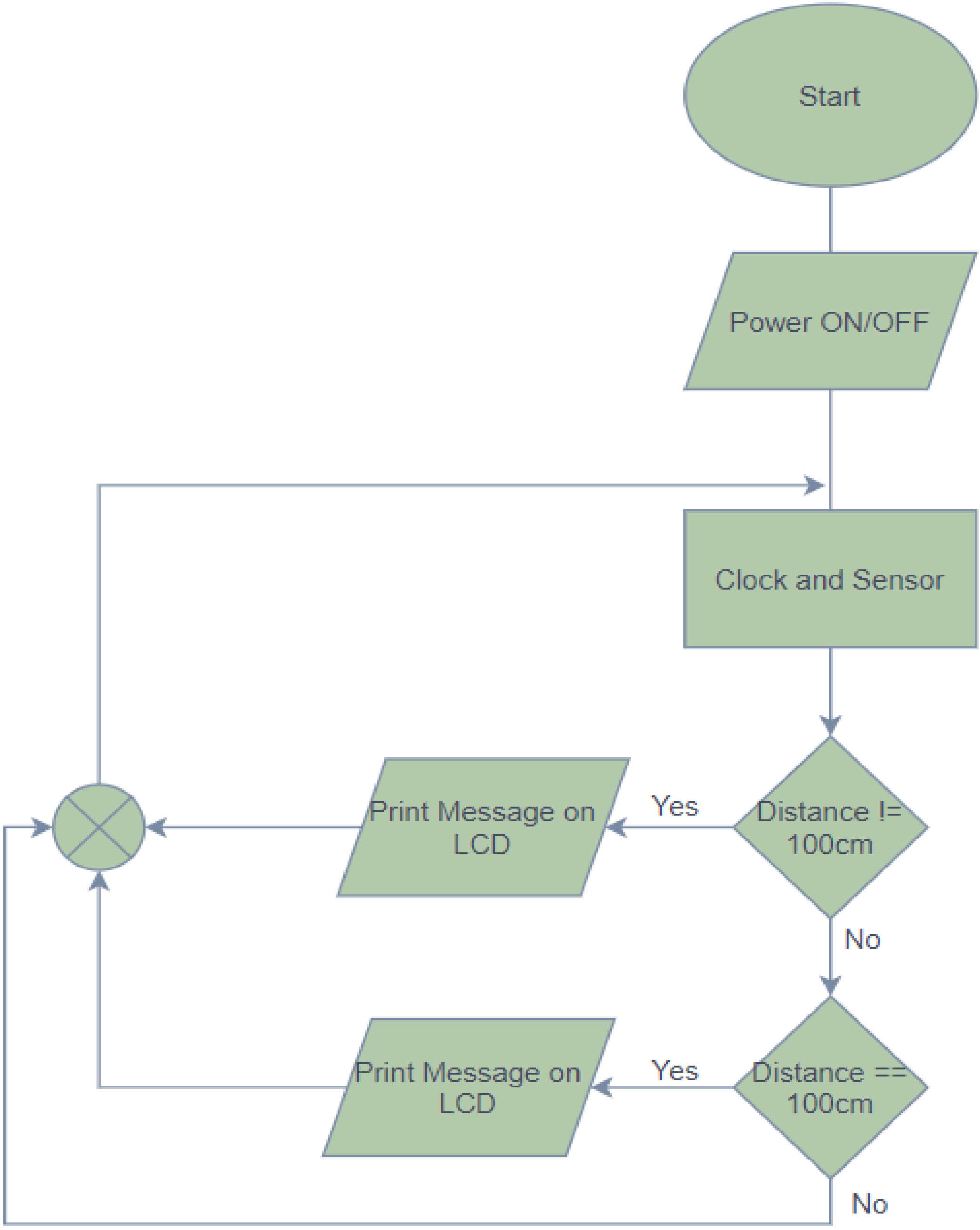
## 4. Implementation details

### a. Circuit description

#### i. Schematic / Logic Diagram



b. **Flow chart**


Start
Power ON/OFF
Clock and Sensor
Distance != 100cm
Yes
Print Message on LCD
No
Distance == 100cm
Yes
Print Message on LCD
No

### c. Program/code/algorithm description

```
#include <avr/io.h>
#include <avr/interrupt.h>
#define F_CPU 16000000UL
#include <util/delay.h>
#include <stdlib.h>
#include <string.h>
#include <avr/lcd4bits.h>
#include <avr/LCD.h>

#define  Trigger_pin          PIND0   /* Trigger pin */
int TimerOverflow

int main(void)
{
        char string[10];
        char Dist[10] = "Distance:", unit[] = "cm";
        char Maintain[10] = "Keep 6ft away", Warning[] = "Mind the Gap";
        long count;
        double distance;
        DDRB = 0xff;
        DDRD = 0x01;
        PORTD = 0xff;
        DDRD |= 1<<PIND5;
        DDRC &=~1<<PINC2;
        sei();                              /* Enable global interrupt */
```

```
TIMSK0 = (1 << TOIE1); /* Enable Timer1 overflow interrupts */
TCCR1A = 0;            /* Set all bit to zero Normal operation */


lcd_init();
while(1)
{
        /* Give 10us trigger pulse on trig. pin to HC-SR04 */
        PORTD |= (1 << Trigger_pin);
        _delay_us(10);
        PORTD &= (~(1 << Trigger_pin));


        TCNT1 = 0;      /* Clear Timer counter */
        TCCR1B = 0x41;/* Capture on rising edge, No prescaler*/
        TIFR0 = 1<<ICF1;        /* Clear ICP flag (Input Capture flag) */
        TIFR0 = 1<<TOV1;        /* Clear Timer Overflow flag */


        /*Calculate width of Echo by Input Capture (ICP) */


        while ((TIFR0 & (1 << ICF1)) == 0);/* Wait for rising edge */
        TCNT1 = 0;      /* Clear Timer counter */
        TCCR1B = 0x01;/* Capture on falling edge, No prescaler */
        TIFR0 = 1<<ICF1;        /* Clear ICP flag (Input Capture flag) */
        TIFR0 = 1<<TOV1;        /* Clear Timer Overflow flag */
        TimerOverflow = 0;/* Clear Timer overflow count */


        while ((TIFR0 & (1 << ICF1)) == 0);/* Wait for falling edge */
        count = ICR1 + (65535 * TimerOverflow);   /* Take count */

        /* 16MHz Timer freq, sound speed =343 m/s */

        distance =(double)count /1000;

        dtostrf(distance, 2, 10, string);/* distance to string */

        strcat(string, unit);           /* Concat unit i.e.cm */

        strcat(Dist, string);

        _delay_ms(200);

        if (distance == 100);

                puts(Dist);

                puts(Maintain);

                _delay_ms(200);

        if (distance != 100);

                puts(Dist);

                puts(Warning);

                _delay_ms(200);

}
```

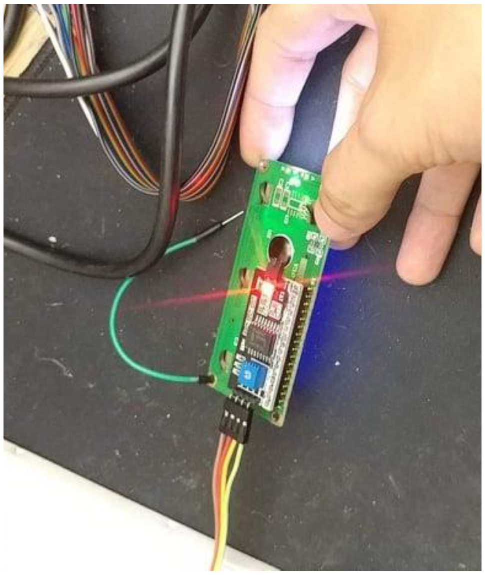
## d. Work assignments of members

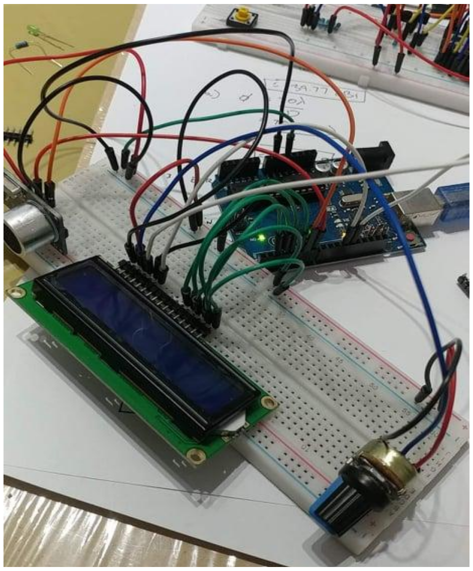
> Karl Escobar – Leader / researcher

> Ken Acula – Rapporteur / researcher

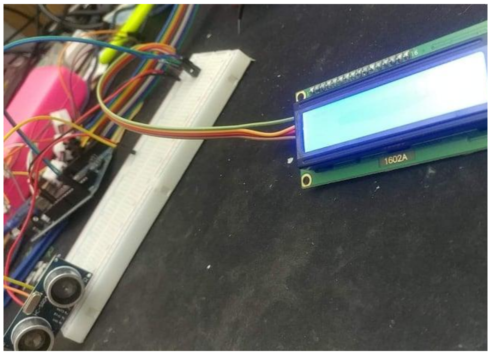
> Angel Lopez - Researcher

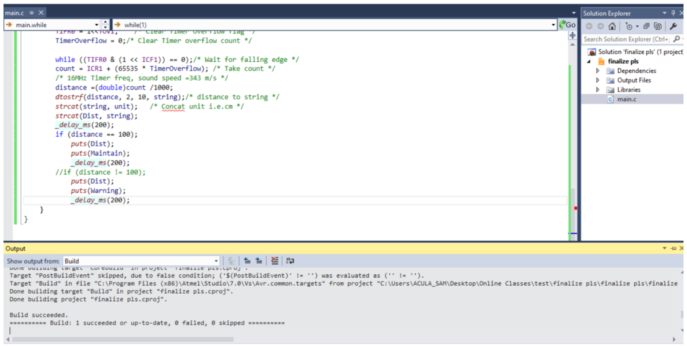
## 5. Test and results





The LCD was bought with a LCM1602 IIC solder at the back of the device. The LCM1602 IIC acts as a display adapter between the LCD and the Atmega328p to communicate the change within the interface. As shown in the photos above, the circuit was based on the previous circuit schematic diagram shown on page 2. The researchers used certain library functions to utilize the LCD in cooperation with the main code.





```
        TIFR0 = 1<<TOV1;    /* Clear Timer overflow flag */
        TimerOverflow = 0;/* Clear Timer overflow count */

        while ((TIFR0 & (1 << ICF1)) == 0);/* Wait for falling edge */
        count = ICR1 + (65535 * TimerOverflow); /* Take count */
        /* 16MHz Timer freq, sound speed =343 m/s */
        distance =(double)count /1000;
        dtostrf(distance, 2, 10, string);/* distance to string */
        strcat(string, unit);   /* Concat unit i.e.cm */
        strcat(Dist, string);
        _delay_ms(200);
        if (distance == 100);
            puts(Dist);
            puts(Maintain);
            _delay_ms(200);
        //if (distance != 100);
            puts(Dist);
            puts(Warning);
            _delay_ms(200);
    }
}
```

```
Done building target "CoreBuild" in project "finalize pls.cproj".
Target "PostBuildEvent" skipped, due to false condition; ('$(PostBuildEvent)' != '') was evaluated as ('' != '').
Target "Build" in file "C:\Program Files (x86)\Atmel\Studio\7.0\Vs\Avr.common.targets" from project "C:\Users\ACULA_SAM\Desktop\Online Classes\test\finalize pls\finalize pls\finalize
Done building target "Build" in project "finalize pls.cproj".
Done building project "finalize pls.cproj".

Build succeeded.
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
```

Every component was tested to see if they were functioning before running the main code on the device. As well as the main code, building the code will prove that the code was reasonable and worked as shown in the pictures above. The documentation for the device was recorded and is available on Google Drive. The maximum amount of distance that the sensors were able to display was up to 100 cm. This was also recorded by slowly increasing the distance between the sensors and an object in front of them.

## 6. Conclusion and recommendation

Customers may pass their time at their leisure and escape the irritation of idle time by adopting the Pandemic Queue Distancing Alert, which allows them to spend their time knowing they are securely socially distant. It works greatly on both structured and unstructured queues. They'll be notified when it's their turn to approach as well. Furthermore, the solutions aid in boosting transparency by providing consumers with exact waiting time forecasts, reducing the discomfort of uncertainty.

The use of queueing solutions has become popular due to the infection prevention standards against COVID, wherein limited customer entry is applied to certain facilities. The rising queueing solutions were kiosk-based and mobile queues, which provide a much more customer-oriented approach and have been implemented in a broad range of variations: In some cases, customers can book time slots online through a web interface or an app, join the queue by using a self-service welcome terminal or ticketing kiosk, queue in on a tablet, or maybe even use a robot assistant on site. It is also possible to simply scan a QR code posted in a shop window. Some customer-based facilities take advantage of Google Lens and their issued QR code to monitor and take information before entering their premises.

## 7. Reference

A. Tom, “Distance measurement using HC-SR04 and ATMEGA328p,” June 17, 2018. [Online]. Available: https://github.com/abtom87/Distance-measurement-?fbclid=IwAR16s-DG4YHXDpj3Ph8klxQ1Mm_T6eRQ3mWTvKSt-6IOodL_kZEe-MuMVPc[Accessed June 15, 2021]

G. Eberhardt, “Queue management systems and their potential in the times of social distancing,” June 18, 2020. [Online]. Available:https://www.ntsretail.com/queue-management-systems-and-their-potential-times-social-distancing [Accessed January 10, 2022]

A. Parajuli, “Distance Measurement using HC-SR04 Ultrasonic Sensor and Arduino,” February 17, 2020. [Online]. Available:https://theiotprojects.com/distance-measurement-using-hc-sr04-ultrasonic-sensor-and-arduino/?fbclid=IwAR17Jtb-aBgdKhcubMq6A-DSKwh2c1CBizrGwMLq7oB0eKZQTNlgZ3yNQuo#source-codesprogram [Accessed May 23, 2022]

J. Picoult, “Managing The Covid Queue: 5 Ways To Keep Customers Happy In The Age Of Long Lines,” July 9, 2020. [Online]. Available:https://www.forbes.com/sites/jonpicoult/2020/07/09/managing-the-covid-queue--5-ways-to-keep-customers-happy-in-the-age-of-long-lines/?sh=25f026682979 [Accessed January 10, 2022]

## 8. Appendix

### a. Bill of materials

- LCD with LCM1602 IIC = Php 300.00
- Arduino Uno = Php 300.00
- HC-SR04 Sensor = Php 160.00