# Project 3: Social Distancing System

**Report Created by:** Kexin Chen, Siquan Wang

**CSE321 Fall 2020**

**Project Due:** 12/10/2020, 11:59 pm

**Project Closes:** 12/11/2020, 11:59pm

# Table of Contents

## Introduction

In today's society, the coronavirus has caused much concern over public health standards. These standards include wearing a mask, limiting the amount one travels, and keeping a minimum distance of at least six feet from other people. It is difficult to gauge whether or not one is at an appropriate distance from another person, so this system aims to help alleviate this problem. This system utilizes an ultrasonic sensor to measure distance and has a buzzer and LCD display to audibly and visibly give feedback based on distance. By eliminating the need to verbally communicate, the spread of germs can be reduced. It also includes a rotary encoder that allows users to adjust the minimum distance desired.

## Features and Specifications

1. Features

    a. NUCLEO-L4R5ZI board
    b. 16x2 1602 LCD display output
    c. Piezo electronic buzzer module
    d. KY-040 Rotary encoder input knob
    e. HC-SR04 Ultrasonic sensor

2. Specifications

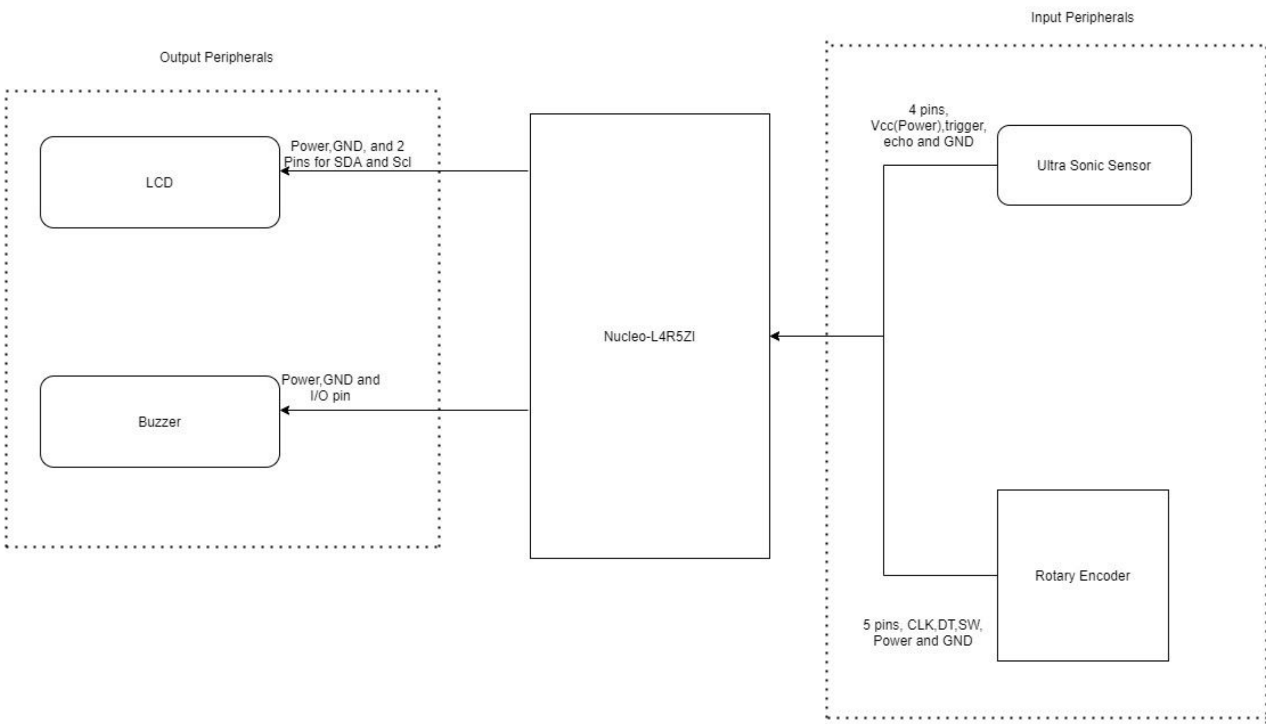
    This distance detecting system includes a buzzer that will make noise and display text on an LCD display warning a person to back up if they are too close to the system. It will also display the person's current distance from the system. When a person is far enough away, as determined by the default distance or one set by the user using the knob, the LCD display will display a default message and show the current distance from the system. By default, the system is set to have a minimum distance of 6 feet, or 183 cm.

    The minimum distance can be adjusted by entering the "Set new distance" menu. This menu is accessed by pressing the on-board user push button. From here, the user can change the minimum distance from around 6 feet to around 13 feet, or 400 cm. This menu is exited by pressing the button again, which will bring the user back to the default screen. The new minimum distance is saved, and the system will act according to this new value.

## Design Process Review

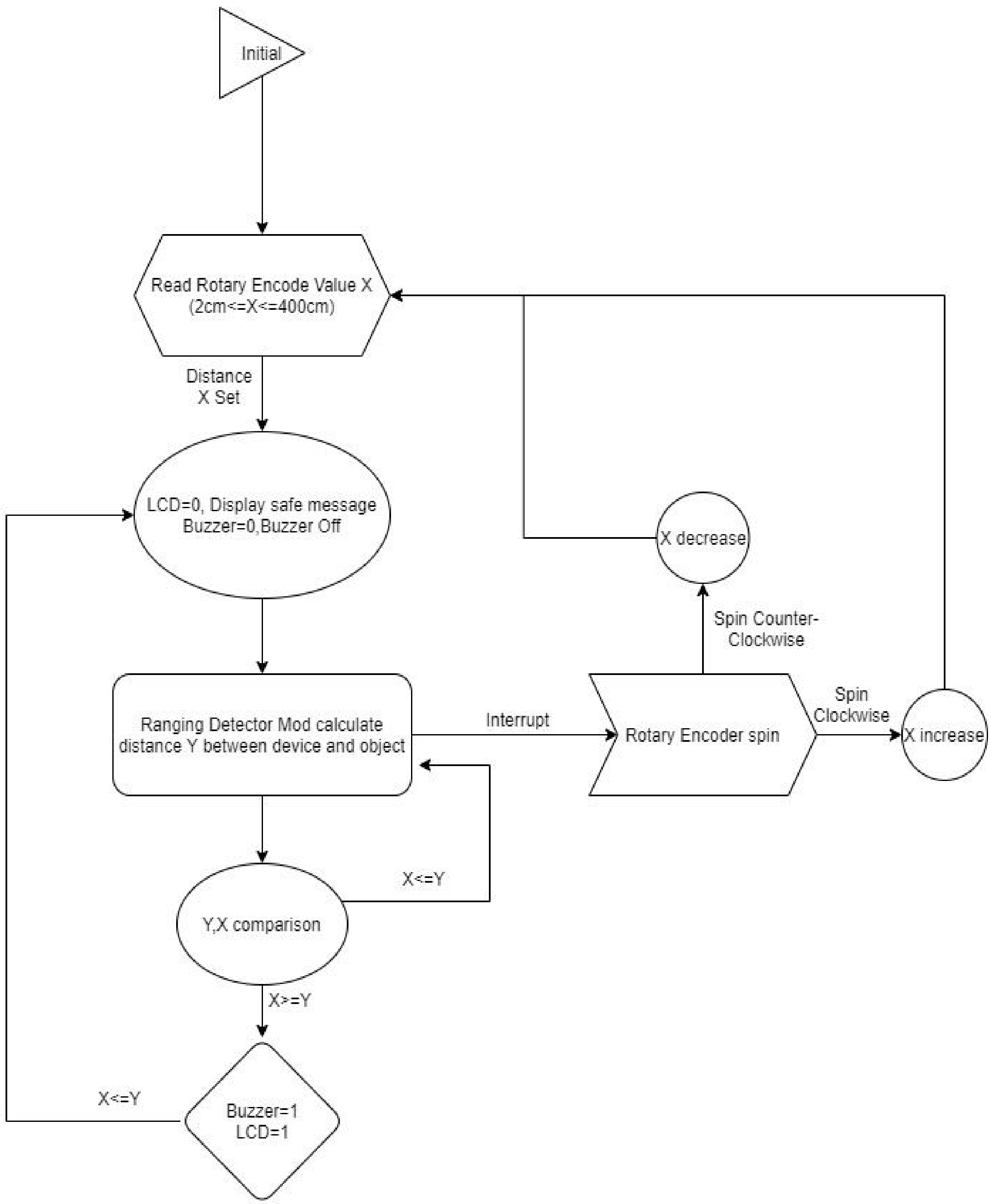
1. Figure out what problem to solve
    a. Coronavirus is a very prevalent issue today. Designing a system to aid in preventing it was determined to be the main idea.
    b. Social distancing is not very well enforced, so enforcing it became an idea
    c. Came up with the idea of creating a system to alert people when they are not at the required distance set by the CDC (6 feet)
2. Decide the inputs and outputs of the system
    a. The peripherals for the system have to ensure that the spread of germs is minimal
        i. This means that no talking should occur and that minimal contact with the system is required
    b. A buzzer and LCD display are perfect outputs, since no talking or touching is required
        i. The buzzer will sound when someone is too close
        ii. The LCD display will show text notifying the user on what to do
    c. An ultrasonic sensor and rotary encoder were chosen as the inputs
        i. The sensor uses waves to measure distance so no contact is needed
        ii. Only one person needs to interact with the encoder to turn the knob, if necessary
3. Consider the constraints and the specifications of the system
    a. Constraints
        i. Sensor can only measure between 2 to 400 cm
        ii. Distance can be adjusted, but only with the rotary encoder
        iii. Ultrasonic sensor cannot have anything blocking its path to object
        iv. Buzzer should be noticeably loud, but not obnoxious
        v. The LCD display should have a response regarding object's distance
    b. Specifications
        i. System should sound the buzzer and display warning on LCD if person is too close
        ii. Distance can be adjusted by the rotary encoder
        iii. System should turn off buzzer and display default message on LCD if person is far enough
        iv. System should default to 6 feet for minimum distance
        v. Distance is measured using the ultrasonic sensor
4. Design the system
    a. Create a flowchart showing the logic of the system
        i. Define some necessary variables
    b. Create a test plan for the entirety of the development process
    c. Gather necessary tools to implement the system

# Block Diagram



## System Diagram

X = Safe Distance, ranges from 183cm(6ft) to 400cm
Y = Distance from sensor, from 2cm to 400cm
LCD = LCD message: 0 = Safe message, 1 = Warning message
Buzzer = Buzzer Status: 0 = Buzzer off, 1 = Buzzer on



## Bill of Materials

1) Cylewet 5Pcs KY-040 Rotary Encoder Module
   - The rotary encoder can count the number of pulse output during rotation in the positive direction and reverse direction through the rotation

2) SainSmart HC-SR04 Ranging Detector Mod Distance Sensor
   - The module automatically sends eight 40 kHz waves and detects whether or not there is a pulse signal back

3) ARCELI 5pcs DC 3.3-5V Passive Low Level Trigger Buzzer Alarm Sound Module
   - Active piezo electronic buzzer module with low level trigger and continuous sound output. It will alarm as long as power is supplied.

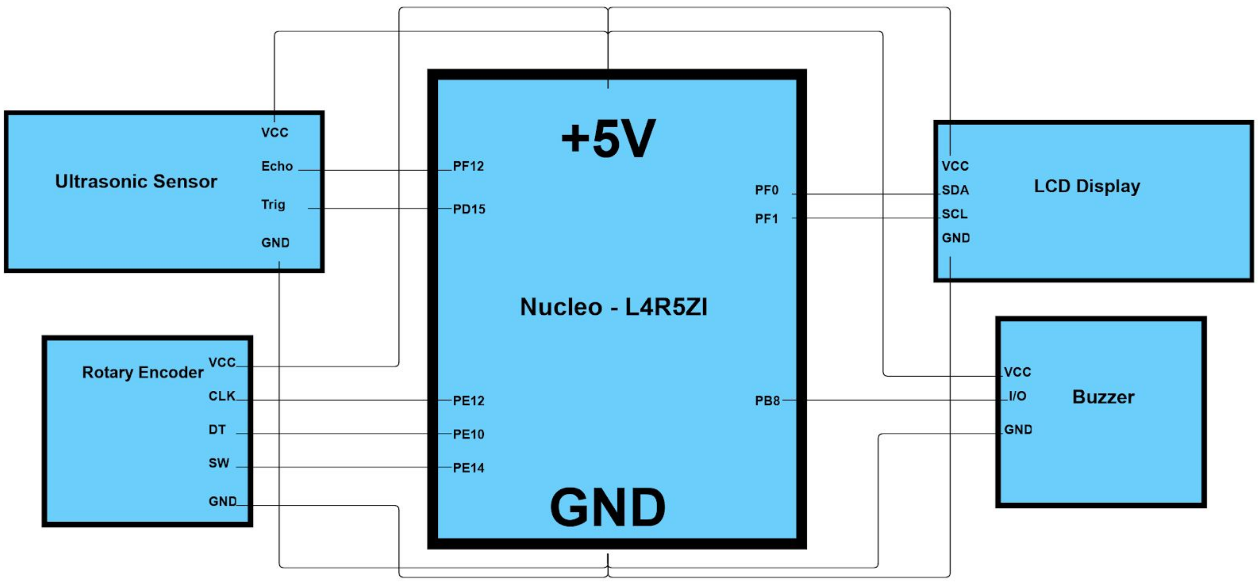
4) HJ Garden Electronic Component Assorted Kit
   - A complete basic kit set for this project. This kit includes a breadboard, various colored LEDs, male to male/female jumper wires, various resistors and capacitors.

5) Nucleo-L4R5ZI Board
   - The STM32 Nucleo-144 board provides an affordable and flexible way for users to try out new concepts and build prototypes by choosing from the various combinations of performance and power consumption features, provided by the STM32 microcontroller.

6) SunFounder IIC I2C TWI 1602 Serial LCD Module Display
   - 16x2 LCD display module supports IIC protocol.

## User Instructions

- Schematic




- Building the system

First, prepare the parts listed in the BOM section, extra sets of male to male and female to male jumper wires are recommended. Assemble the system with the parts according to the schematic provided above. It is recommended to set up the ultrasonic sensor on the breadboard and then place the breadboard on a flat surface for a constant, accurate measurement. After assembling the system, download the project code with the supporting files and run it through Mbed studio. Before initializing the system, make sure to face the ultrasonic sensor towards an empty, unobstructed area.

- How to use the system

Power on the system, make sure the sensor has a clear path to whatever it is detecting. After one short buzzer beep, the system is initialized. Press the user button on the Nucleo board to enter the "Set new distance" menu mode. This menu allows users to set the desired safe distance beyond the recommended 6 ft (183 cm) distance set by the CDC. Any value lower than 6 ft, down to 1 ft (31 cm) should only be used for testing purposes. Turning the rotary encoder in this menu can adjust the safe distance. Turning clockwise will increase the safe distance by 1 cm and turning counterclockwise will decrease it by 1cm. Press the user button again to exit the menu and to set the desired safe distance. Idling in the "Set new distance" menu for more than 30 seconds will reset the system with the default minimum distance of 183 cm. This system will run forever in the default state.

## Test Plan

Before implementation:

- Test all of the components to make sure they function properly
    - Ultrasonic sensor is able to detect objects in range
    - Rotary encoder changes value based on rotation
    - LCD lights up and displays text
    - Buzzer is able to make sound
- Check the system information to ensure that the NUCLEO board functions correctly
    - Use *ram_size, rom_size, and cpu_id* to verify that RAM and ROM are the appropriate sizes and that the correct CPU ID is returned

During implementation:

- Make sure the ultrasonic sensor is returning the correct distance
- Check that the rotary encoder changes the safe distance appropriately
    - Increases it when turned clockwise
    - Decreases it when turn counterclockwise
- Have the LCD display the current distance from an object
    - If current distance is less than safe distance, display warning
    - If distance is greater than/equal to safe distance, display current distance
- Buzzer should make sound based on distance
    - If distance is less than safe distance, make sound
    - If distance is greater than/equal to safe distance, no sound
- Check to make sure that errors do not occur with the memory
    - Using *current_size* and *alloc_fail_cnt* to verify the heap size at certain points and that no allocations have failed

After implementation:

- System should start up with displaying a message " Social Distance" to the first line and current distance from a detected object to the second line on the LCD
- Upon button press, if the rotary encoder is turned, the LCD shows the changing safe distance value when in the "Set new distance" menu
    - Next button press returns user to default menu, knob should not do anything now
- When a person is within the range of the ultrasonic sensor…
    - LCD displays warning and Buzzer sounds off when distance is less than safe distance
    - LCD displays the safe distance and Buzzer is silent when distance is greater than/equal to the safe distance
- Check the CPU for runtime and idle time
    - Using *uptime* and *idle_time* to check that the system is not stuck idling for too long, since it should always be actively doing something

## Development Timeline

| Date | Revision | Changes |
|---|---|---|
| 10/31 | Stage 1 Close Out | Project group formation.<br>Created group Github repository.<br>Basic coding template created.<br>Initial definition of system<br>Project initial start. |
| 11/1 | Stage 2 Close Out | Creation of social distance system statement for stage submission.<br>Initial constraints and specification for project defined. |
| 11/09 | Stage 3 Close Out | Project parts listed and ordered.<br>Initial design for a social distance system created.<br>Searching for part datasheets. |
| 11/16 | Stage 4 Close Out | Project parts testing, all parts are functional.<br>Start searching for references for ultrasonic sensors.<br>Interrupt callback functions prototype added.<br>LCD display set up.<br>Uploaded datasheets for parts to Github |
| 11/22 | Stage 5 Close Out | Added ultrasonic function to the main file,<br>Change LCD display message.<br>Testing buzzer function. |
| 11/24 | Final edition 1 | Adjusted ultrasonic sensor check time, LCD now |

| | | |
|---|---|---|
| | | display detected object distance correctly.<br>Buzzer function added, buzzer now works when the board user button is pressed. |
| 11/27 | Final Edition 1 | Watchdog fixed.<br>Buzzer now works as intended (object within safe distance).<br>Testing encoder libraries. |
| 12/6 | Final Edition 1 | Added encoder function.<br>Modified rotary encoder library to run on Mbed.<br>Added rotary encoder library QEI. |
| 12/7 | Final Edition 1 | Fixed encoder function, encoder now adjusts min. distance correctly,<br>Changed user button 1 to change between modes.<br>Fixed user button bounce.<br>Fixed encoder rotation bounce.<br>Cleaned up code. |
| 12/8 | Final Edition 1 | Fixed minor bugs.<br>Formatted code, added comments and pushed the final main.cpp on github. |
| 12/9 | Final Edition 2 | Changed 3 variables to be volatile and added a mutex to meet synchronization requirement for the project. |
| 10-Dec-2020 | Final Submission Deadline | Submitted final code. |

## Task and Contribution Breakdown

| Kexin Chen | Siquan Wang |
|---|---|
| ● Configured the output devices<br>○ Set up and completed functions for the LCD and Buzzer<br>○ Found corresponding libraries and figured out how to use them in relation to the project.<br>● Configured Watchdog<br>● LCD displays distance measured from ultrasonic sensor<br>● Buzzer turns on and off based on distance<br>● Found QEI library for the Rotary Encoder<br>● Fixed menu switching with button<br>● Added synchronization<br>● Cleaned-up code and add comments | ● Configured the input devices<br>○ Set up and completed functions for the ultrasonic sensor and rotary encoder<br>○ Found reference/libraries and testing the logic on parts, configure the ultrasonic sensor the return correct distance in cm<br>● Configure user button that switch between detecting mode and menu mode<br>● Debounce rotary encoder inputs<br>● Debounce user button<br>● Update the QEI library to work on current Mbed version<br>● Configure the ultrasonic sensor stop working when system is in menu mode |

## Future Design Considerations

### Shortfalls for System Design

One of the biggest shortfalls of the design of the system is that it requires a clear path to detect if a person is far enough away. The system will detect anything that is in range, meaning that it will detect surrounding objects rather than just a person if the object is closer to the sensor. This is due to how ultrasonic sensors work, as they measure distance using waves and their reflections.

This system is also not able to be used in a wide range of applications and is essentially immovable. It is best used in places that people have to line up and only one person is detected at a time. This means that in crowded areas the system will most likely not work as intended as people may not know if they personally are too close or not.

### Communication Feature

An UART input peripheral device that allows users to input the safe distance with keys, like a keyboard, can be a great replacement for the rotary encoder. The rotary encoder on the prototype now can only provide either accurate, slow inputs or approximated, fast inputs to modify the value of safe distance. This is because one knot rotation only changes the safe distance by a set amount. A keypad input can solve this problem by providing both accurate and fast inputs simultaneously.

### Memory Management

Currently in the program, a majority of the variables are stored as global variables that are used throughout the program. This makes it so that they are only allocated for once. An EventQueue was used to handle the interrupts that occur. The EventQueue is especially large since it allocates the max size of an event times 32. Space could most certainly be saved if the EventQueue was smaller.

A different way to improve the system is to use memory allocation functions. Using a function like realloc() would improve the system by decreasing the amount of allocations and deallocations needed during execution, therefore increasing its speed and efficiency. There are a few temporary variables that are the same size or smaller, so it is possible to reuse the same pointer and memory chunk by utilizating realloc(). This is because realloc() reuses the same memory as long as the existing chunk is large enough.

## Direct Memory Access

A current connection that will benefit from a DMA integration is the connection with the LCD display. The code currently calls many functions for the LCD to print, clear, and position its cursor, all of which take a decent amount of time to complete. By using DMA with the LCD, the LCD can more quickly update its display since there is now a direct connection from memory to the LCD. This also has an effect on the rest of the system, as the CPU no longer has to focus on outputting data for the LCD first before moving on to other tasks, increasing the overall speed and efficiency of the program.

## General Improvement

Adding a user setting for buzzer volume is one of the major features that should be added to the system. The current buzzer can only buzz at a constant high pitch and volume tone which can be very annoying after a while.

Designing a container to hold the components is another way to improve the system. Holding the components in fixed positions allows convenient carry for users. It also can improve the accuracy of distance measurement.