Food Waste Reduction Inventory Notifier

CSE321 Fall 2021: Project 3

Misha Nelyubov

December 3, 2021

Misha Nelyubov
CSE 321 Project 3
Fall 2021

Table of Contents

01. Introduction

This system aims to assist in reducing waste of short lifespan store foods by tracking how much of a container's inventory is occupied over the course of a work day. If there is food remaining in the container at the end of the work day, the system will generate a sound to alert workers that the remaining food in the container can be taken home and consumed before it goes bad.

02. Specifications and Features

a) Features

- Once powered on and configured, the system will remember the current real-world time for the duration of the power cycle, or until modified.
- A custom 24-hour "closing time" can be configured down to the second to meet the exact needs of its specific implementation.
- Any values not input during time configuration will default to 0 to save input time.
- The distance sensor regularly polls to check how much of the container is occupied and provides a real-time percentage estimate once calibration is completed.
- The points at which the container is considered empty and full can be calibrated to support a wide range of containers.
- Once configured, the system alarm can be toggled on and off by pressing the "#" key on the matrix keypad.
- The buzzer will play a series of tones to alert surrounding workers when the container is NOT empty after "closing time" has been passed.
- At any point, the key "D" can be pressed to reconfigure and recalibrate the system.

b) Specifications

- The distance sensor has a maximum effective range of 400 cm (SparkFun Electronics).
- The distance sensor has a minimum effective range of 2 cm (SparkFun Electronics).
- The distance sensor is polled for a new distance value once every 100ms (10 Hz).
- The displayed value for the distance sensor is always the average of the last four distance sensor samples that were taken within the effective sensor range.
- The text displayed on the LCD is refreshed for updates every 100ms (10 Hz).
- Current Time and Closing Time are both 24-hour time values with valid values ranging from 00:00:00 up to 23:59:59.
- The matrix keypad is polled at 10ms per column. Polling is stopped while any matrix key is pressed down.

- If a matrix key is perceived by the system to be pressed down for 30 consecutive seconds, the system watchdog will assume that the key has gotten stuck and force a system reset.

03. Integration of Required Features

a) Watchdog Integration

The system watchdog was implemented to protect users from a keypad input lockout that would risk the system being unresponsive to further user keypad input. This protection point is considered crucial because loss of access to the keypad prevents any user from being able to modify the state of the system, including loss of the ability to directly disable an ongoing alarm. Because user key inputs are tracked based on rising and falling edges of button presses with an internal variable storing the currently pressed key, the watchdog timer is allowed to count down the 30 seconds whenever any key is pressed. If no key press is detected for any 1ms during that countdown, this indicates that a user is able to press an input key again and the watchdog reset will be called off. The watchdog is periodically reset back to counting down from 30 seconds while no key is pressed.

b) Synchronization Integration

Synchronization is necessary to ensure that data flow between input and output peripheral control is not accessed while it is being updated and that system resources are used efficiently. To ensure efficient and cooperative operation, a mutex is used to control synchronization for modifying output peripherals. Anytime that a change is made in the system that would affect the state of the output peripherals, a mutex-protected flag is raised indicating that output changes have been made. The next time that the function that updates the state of the output peripherals is called, it will check if that flag is raised to be informed if any action needs to be taken. If that flag is raised, the output peripherals will be modified. Otherwise, the function exits immediately to save system resources.

c) Bitwise Driver Control Integration

A bitwise output driver was implemented for the distance sensor trigger. Modification of the output bit from a value of 0 to a value of 1 for a duration of 10 microseconds before returning to 0 is responsible for sending a trigger signal to the distance sensor in order to cause the sensor to generate a distance measurement (SparkFun Electronics). Configuration of the bitwise driver occur in the main function. The main function is responsible for enabling port C and configuring pin C9 as an output to be used by the distance sensor. Modifications to the output value of the pin are made by the function pollDistanceSensor.

d) Critical Section Protection

Critical Section Protection is necessary when more than one thread needs to access to a resource that is accessed by more than one thread. Mutual exclusion is implemented to prevent race conditions where more than one thread attempts to access the same resource, resulting in an unintended system state. A Mutex lock encapsulates all instances of functions reading from and writing into the LCD output text table. If the mutex is not used, a thread writing to the output table might result in a partially changed value being set into the string before a context switch causes that partially changed value to be printed onto the LCD before the completed new data is written. Usage of the mutex ensures that any thread attempting to access the LCD output table must wait until no other thread is accessing the table before being granted access to proceed with its operation.

e) Multithread Implementation

Individual threads are implemented in the system to control the multiple system peripherals. All operations pertaining to the distance sensor are executed by the Distance Sensor Thread. This ensures that all internal variables associated with the distance sensor are only modified by whatever function is being executed by that thread. All non-ISR functions that handle the behavior of the distance sensor are called by being put into the distance sensor event queue or by being called by a function being executed from that event queue. Use of primarily thread-internal variables that do not need to be locked by mutexes minimizes the lockout period that other threads may experience while the Distance Sensor Thread is modifying shared system variables. All threads and tickers to enqueue periodic events for the threads are started by the main function after interrupts and peripheral drivers have been configured. Function execution and data access by all implemented threads is visualized in Section 05. Block Diagram.

f) Interrupt Implementation

Interrupts are used to immediately react to changes in the state of the echo signal sent from the distance sensor to the NUCLEO. The alternative method to using interrupts to acquire the echo signal data would be polling the input pins. Due to an accurate count of milliseconds between the rising and falling edge sensor echo signals being essential for accurately calculating the current distance, the low cost and high precision solution of interrupts is optimal. The interrupts are declared in a global scope and initialized within the main function to call two ISR functions, corresponding to a rising and falling edge trigger event. These functions record the time in milliseconds between the distance sensor being polled and the edge event being detected. The falling edge ISR enqueues a non-ISR function to be handled by the distance sensor thread to calculate the data rapidly recorded during the short interrupt handler periods.

04. Design Process

a) Ask

i) Purpose

In an effort to minimize food waste, foods that aren't sold at the end of a work day can be taken home by the workers and consumed while the food is still in good condition. Design a system that will alert staff members at the end of a work day if there is still food in a container that can be taken home for consumption.

ii) Inputs

- Distance Sensor
  - Measures the amount of contents in the food container.
- Matrix Keypad
  - User Interface to configure time and calibrate system.

iii) Outputs

- Alarm Buzzer
  - Creates a sound to alert surrounding workers if there is food in the container after closing time.
- LCD
  - User Interface to configure time, calibrate the system, and display current time and container usage.

iv) Constraints

- This system is limited to monitoring the space usage of a single container.
- The container that is being observed must not have any object, including any transparent layer such as glass, between the distance sensor and the measuring point of the container.
- The system will need to be informed of the time of day, after what point in time the alarm should sound, what distance indicates an empty container, and a second reference distance to indicate a full container.
- The dimensions of the product inside of the container may limit the effective detection range or accuracy of the distance sensor.

b) Research/Imagine

In order to recognize if there is food remaining that can be taken home, a distance sensor can be used to approximate how much food is currently in a container. For this approximation to be accurate, the distance sensor's data will also need reference distances for the maximum length away from the sensor when the container is empty and the minimum length away from the sensor as a reference for when the container is full. One button will be needed to allow a user to confirm these distances.

In order to recognize when workers should be alerted that there is remaining food, the system will need to be aware of the current time and the time after which it should notify workers of any remaining contents. Ten additional buttons will be needed to allow an easily accessible range of input numbers.

For both of the required user inputs, an output indicator of what the system's user has configured and is about to confirm would improve the ease of use of the system. An LCD can be used to display the current time, closing time, and distance between the sensor and the nearest object to it in the container.

The system should notify workers in such a way that they do not need to be actively monitoring the system in order to be alerted if it is past closing time and the container is not empty. A buzzer module would be able to accomplish this by creating a sound that will alert everyone in the immediate area. This buzzer-produced sound should be designed to not be excessively unpleasant nor cause any hearing problems. An additional input button should be dedicated to explicitly shutting down/restarting the alarm.

05. Block Diagram

The following color code is used in the block diagrams for each thread of the program listed below:

Block Color	Meaning
Red	This block represents a ticker.
Orange	This block represents a thread.
Yellow	This block represents an ISR function.
Green	This block represents a non-ISR function.
Half - Green	This block represents an Event Queue of non-ISR functions.
Blue	This block represents a direct pin or API access to hardware peripherals.
Purple	This block represents data access to internal variables. Parenthesized numbers that precede variable names indicate that the variable is protected by an ordered Mutex lock.

Figure 1: Main program execution behavior

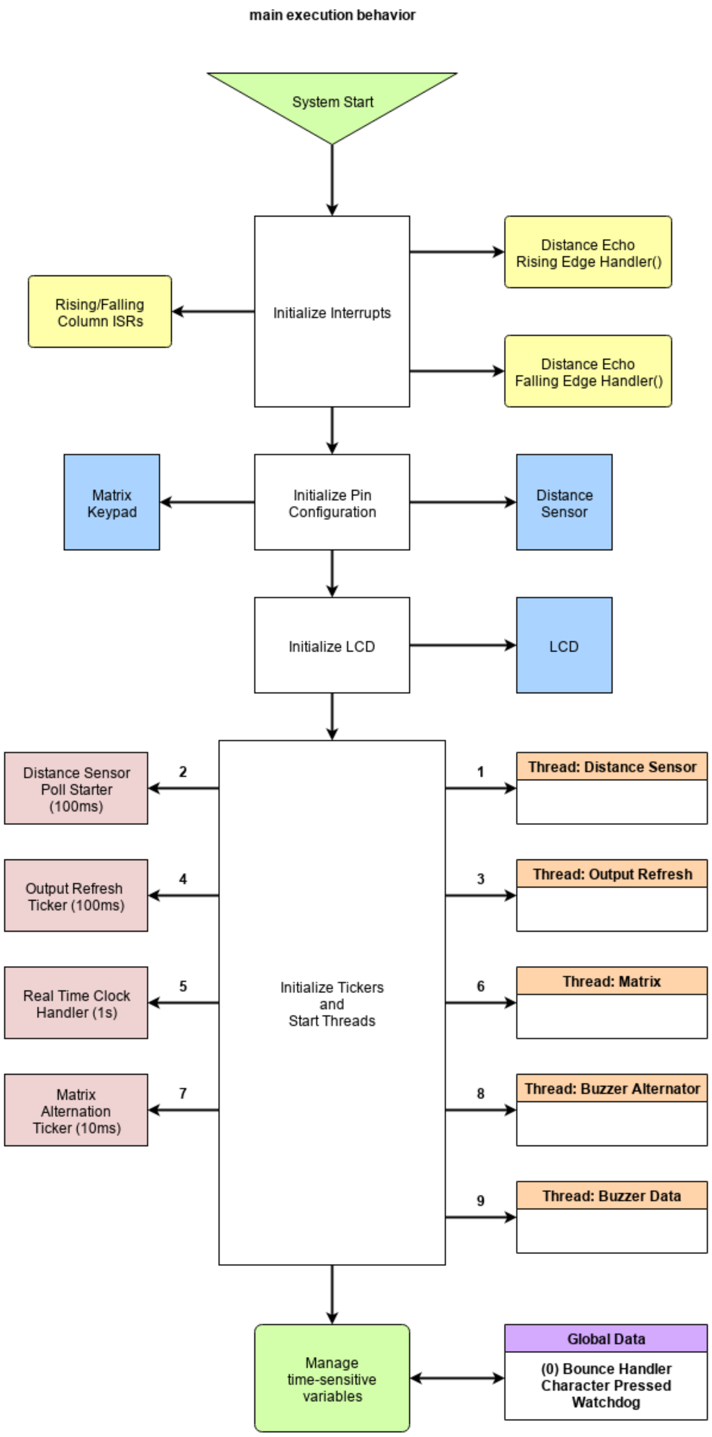


Figure 2: Distance Sensor Thread

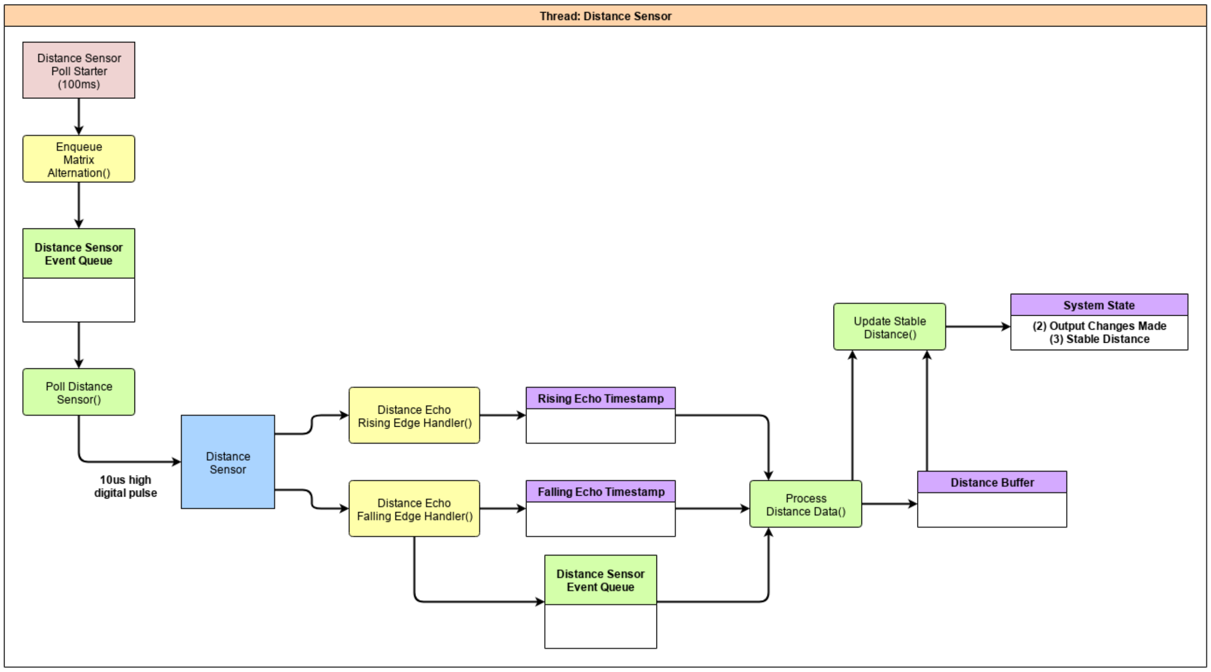


Figure 3: Matrix Input Thread

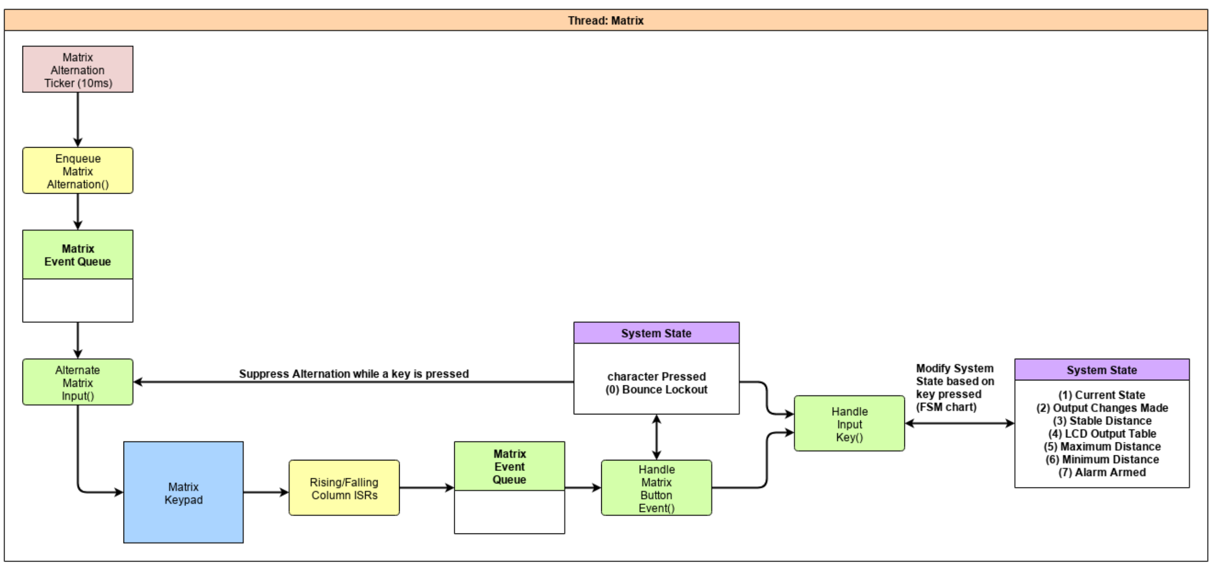


Figure 4: Output Refresh Thread

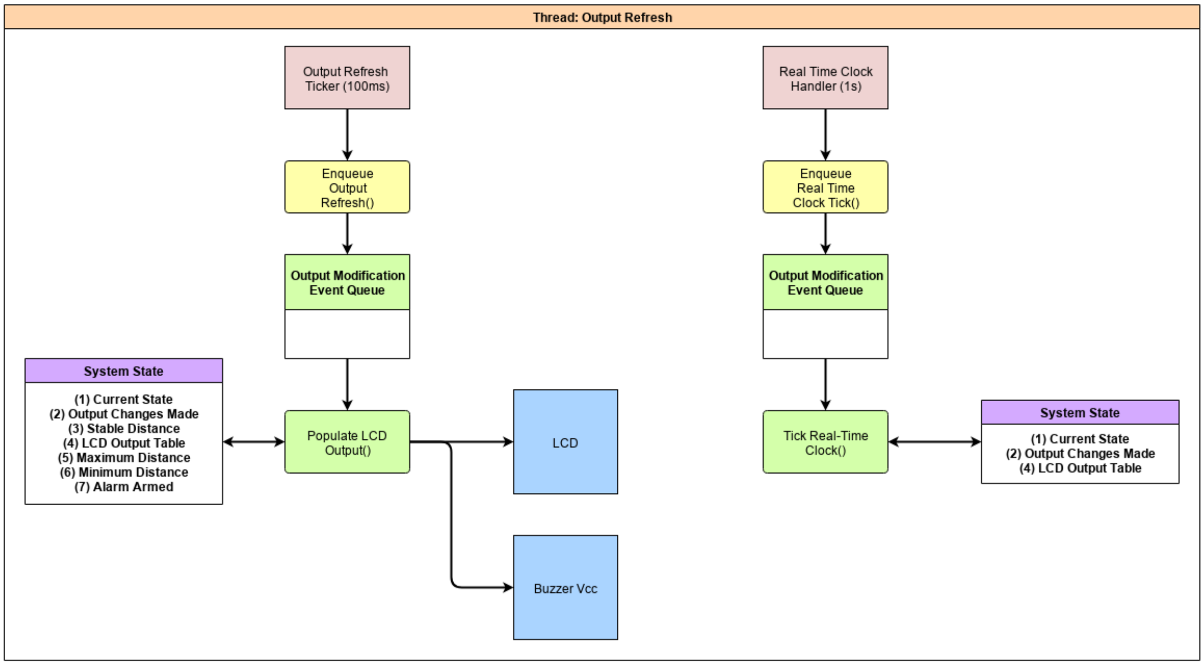
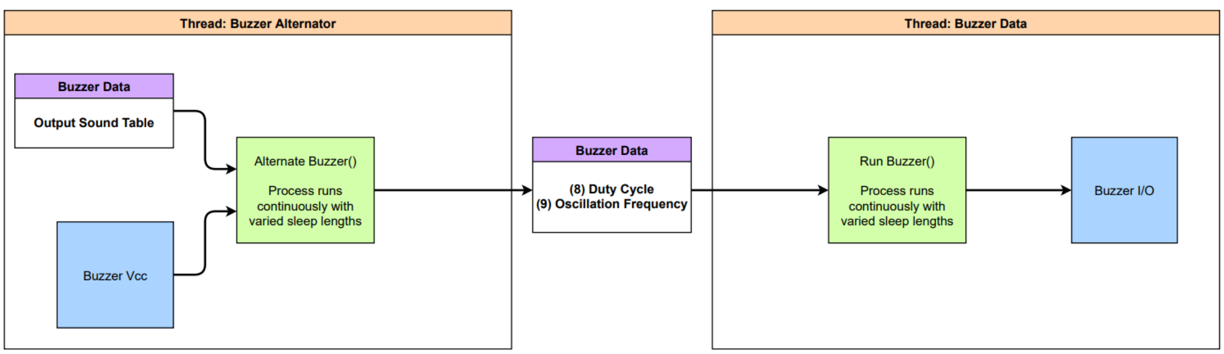


Figure 5: Buzzer Threads



06. State Diagram

The following diagram covers the conditions necessary for the transitions between states.

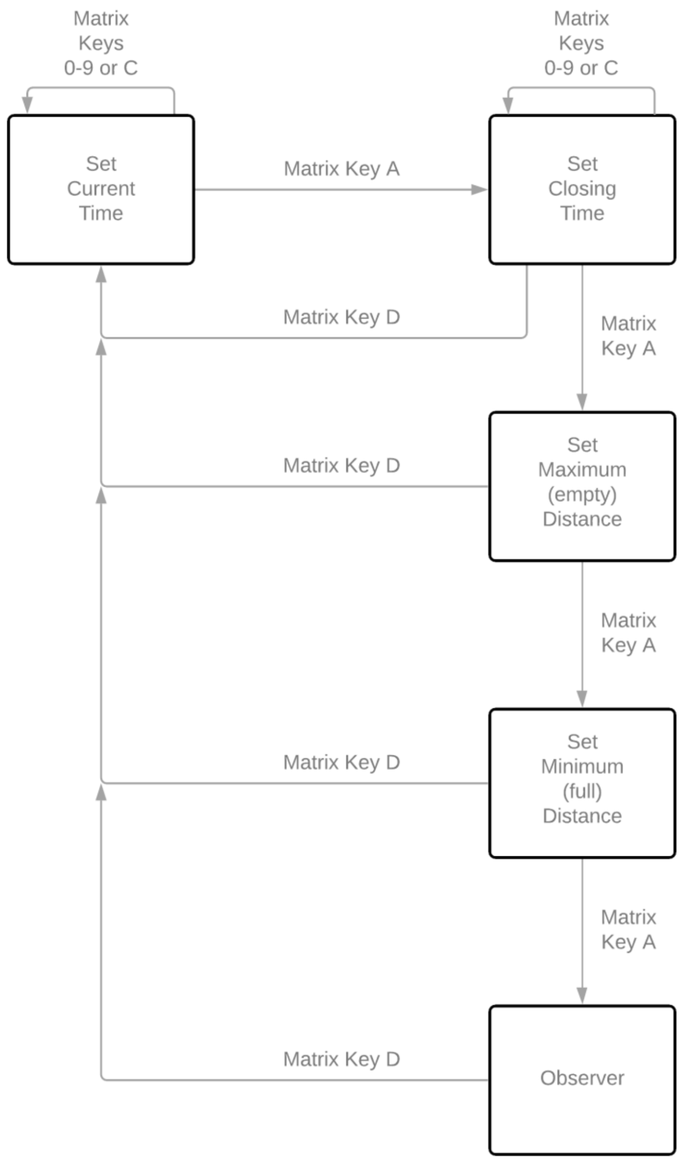
The matrix key “A” is used as an option during all configuration stages to proceed to the next state.

The matrix key “C” is used in the time input stages to reset the input time to “hh:mm:ss” and reset the user data input cursor to the tens of hours position.

The matrix key “D” is used as a universal reset to the “Set Current Time” state in order to modify the system configuration.

The system buzzer peripheral will only produce sound when the system is in the Observer state.

Figure 6: Matrix Input State Diagram



07. Bill of Materials

The following hardware will be required to create the system:

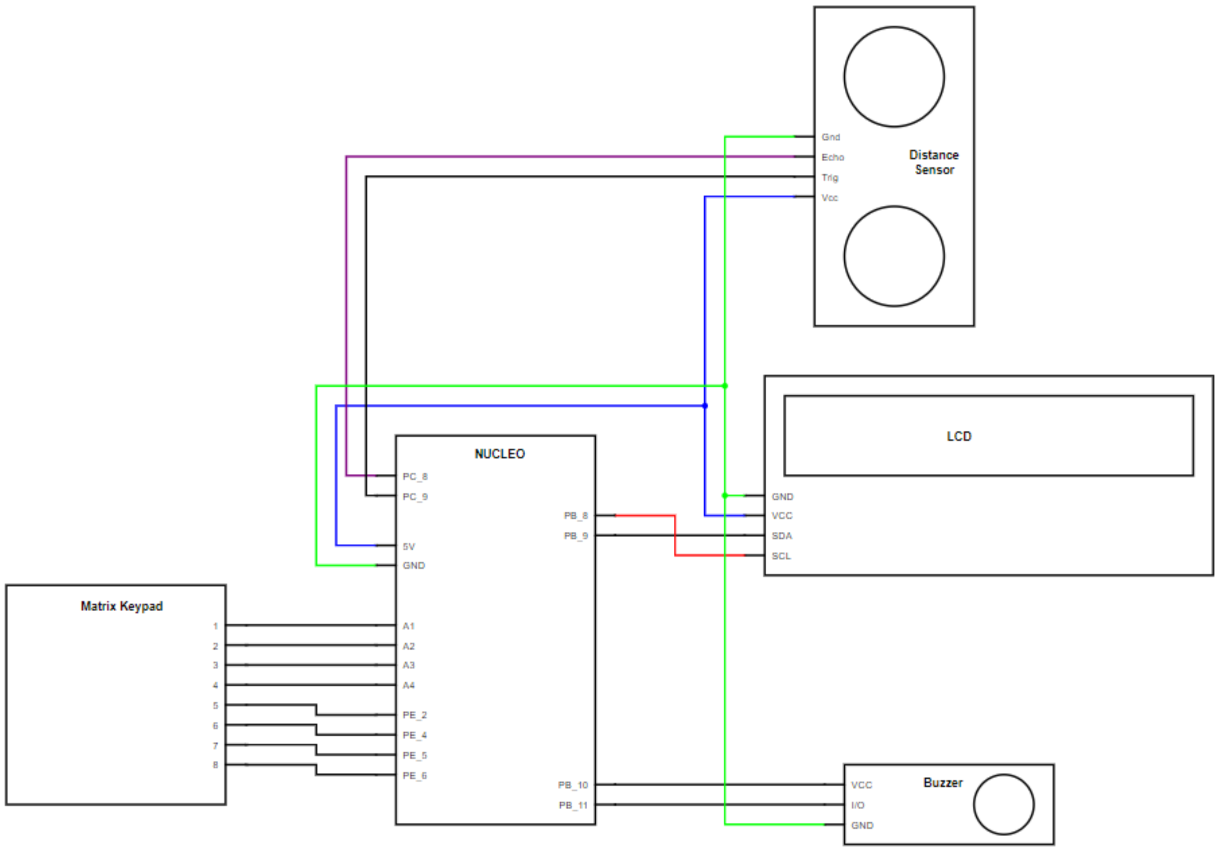
- NUCLEO L4R5ZI microcontroller
- 4x4 matrix keypad (8-pin)
- JHD1804 LCD
- 4-pin ribbon cable for connecting LCD to breadboard
- HC-SR04 Distance Sensor
- MH-FMD Buzzer Module (Low Level Trigger)
- Solderless breadboard
- USB 2.0 A to USB 2.0 Micro B cable
- Jumper wires (no less than 21)

08. Instructions

a) Schematic

The following set of connections between the NUCLEO and peripherals need to be established (USB connection to a computer not pictured):

Figure 7: System wiring Schematic



b) Construction Instructions

In order to build the system, the following connections need to be established between the NUCLEO and peripherals using jumper wires. Use of a solderless breadboard and side rails for 5V (VCC) and GND is recommended.

Matrix Keypad:

- Matrix pin 1 must be connected to the NUCLEO Pin A1 (PC_0)
- Matrix pin 2 must be connected to the NUCLEO Pin A2 (PC_3)
- Matrix pin 3 must be connected to the NUCLEO Pin A3 (PC_1)
- Matrix pin 4 must be connected to the NUCLEO Pin A4 (PC_4)
- Matrix pin 5 must be connected to the NUCLEO Pin PE_2
- Matrix pin 6 must be connected to the NUCLEO Pin PE_4
- Matrix pin 7 must be connected to the NUCLEO Pin PE_5
- Matrix pin 8 must be connected to the NUCLEO Pin PE_6

LCD:

- The LCD pin labeled “GND” must be connected to the NUCLEO pin GND
- The LCD pin labeled “VCC” must be connected to the NUCLEO pin 5V
- The LCD pin labeled “SDA” must be connected to the NUCLEO pin PB_9
- The LCD pin labeled “SCL” must be connected to the NUCLEO pin PB_8

Distance Sensor

- The Distance Sensor pin labeled “Gnd” must be connected to the NUCLEO pin GND
- The Distance Sensor pin labeled “Echo” must be connected to the NUCLEO pin PC_8
- The Distance Sensor pin labeled “Trig” must be connected to the NUCLEO pin PC_9
- The Distance Sensor pin labeled “Vcc” must be connected to the NUCLEO pin 5V

Buzzer

- The Buzzer pin labeled “VCC” must be connected to the NUCLEO pin PB_10
- The Buzzer pin labeled “I/O” must be connected to the NUCLEO pin PB_11
- The Buzzer pin labeled “GND” must be connected to the NUCLEO pin GND

Once all connections between the NUCLEO and peripherals have been established, connect the NUCLEO USB-B port labeled “USB PWR” to the computer that will load the program onto the NUCLEO. This computer will need to have MBED Studio installed to compile the code and load it onto the NUCLEO.

After MBED Studio is installed, perform the following steps to load the necessary program onto the NUCLEO.

1. Download the GitHub repository containing the necessary program. If Git is installed on the system, this can be done with the following command:
    a. git clone https://github.com/CSE321-Fall2021/cse321-portfolio-MSNelyubov.git
2. In MBED Studio, click on File > Open Workspace.
3. Navigate to the path where the repository was downloaded.
4. Click on Project 3 and then click *Select Folder*.
5. Set the *Active program* to “Project 3”.
6. Set the *Target* to “NUCLEO-L4R5ZI …”.
7. Click on the play button to run the program.

c) Usage Instructions

i) Powering on the System

1. Turn on the system by connecting a USB cable from the NUCLEO to a 5V power supply.
    a. The text "Set current time" and "(24hr) hh:mm:ss" should appear on the LCD.

ii) Configuring System Time

2. Input the current time into the system using the matrix keypad. All system times are on a 24-hour clock. Input time begins on the left in the tens of hours digit and moves right until the seconds digit is entered.
    a. E.g. if the current time is 1:24:56 PM, the sequences of button presses would be “1”, ”3”, ”2”, “4”, “5”, ”6”.
    b. If any digit is not entered, that time position will default to 0. E.g. “09:mm:ss” will be converted to “09:00:00”.

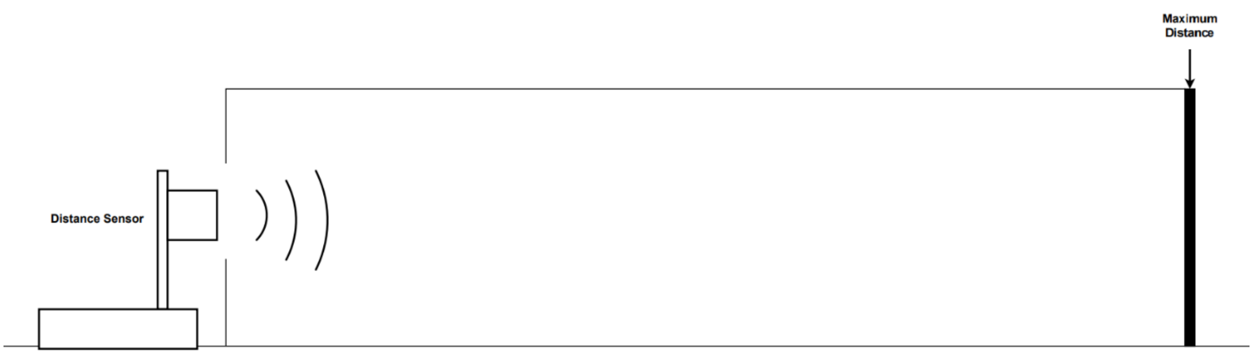
c. If the time was input incorrectly, press “C” on the matrix keypad to clear the currently input time and restart entering the time from the tens of hours.

3. Press “A” on the matrix keypad to confirm the current time.
4. At the “Set closing time” screen, set the time after which the system will activate the alarm if objects are seen in the container.
   a. The same time input convention is followed here as in step 2.
5. Press “A” on the matrix keypad to confirm the current time.

iii) Calibrating Container Distances

6. At the “Set empty” screen, move the system so that the distance sensor accurately measures the distance in centimeters to the far end of the container that the system should monitor.
   a. The distance sensor has an upper distance limit of 400 cm (SparkFun Electronics).
   b. A direct path with no barriers (including transparent film, plastic, and glass), is necessary between the sensor and far end of the container.
   c. To maximize the accuracy of the measurement, it is recommended that the container is emptied for calibration.
   d. It is recommended to have the system at the back or highest elevation point of the container so that the most accurate measure of capacity can be provided.
   e. The following graphic shows how the distance sensor should be positioned with respect to the container.

Figure 8: Positioning the distance sensor with respect to the container that it is observing.



7. Once an accurate measurement of the distance between the distance sensor and the back of the container is displayed on the LCD, press “A” on the matrix keypad to confirm this distance.
8. At the “Set full” screen, place an object inside of the container as close to the distance sensor as possible. This will be used to mark off when the container is 100% full.
   a. The distance sensor has a lower limit distance of 2 cm (SparkFun Electronics).
   b. If the closest that an object can go toward the sensor is less than 2cm, anomalous sensor readings may be produced. Move the object away from the sensor so that a low distance of 2~5 cm is displayed.
9. Once an accurate measurement of the distance between the distance sensor and the closest point within the container to the distance sensor is displayed on the LCD, press “A” on the matrix keypad to confirm this distance.

iv) Actions During Active Observation

10. The system is now in a state of actively observing the capacity of the container.

11. If any configuration or calibration step needs to be modified, press "D" on the matrix keypad to return to the beginning of the system configuration and proceed from step 2.
12. To toggle the system alarm on/off, press "#" on the matrix keypad.
    a. A single "#" symbol will appear on the LCD when the alarm is armed and disappear when the alarm is not armed.
13. Store food items in the container as normal and wait for the current time to go past the programmed closing time.
14. If the alarm goes off, press "#" on the matrix keypad to silence the alarm, and then remove any items in the container to be taken home for consumption if they are still in good condition.

09. Test Plan Instructions

a) Buzzer Sound Testing

1. Turn on the system by connecting a USB cable from the NUCLEO to a 5V power supply. The following text should appear on the LCD:
    a. `"Set current time"`
    b. `"(24hr)  hh:mm:ss"`
    c. If the LCD does not display a backlight, the LCD may not have been properly connected to the 5V and GND pins.
    d. If this text does not appear, the LCD may not have been properly connected. Restart the system by pressing the on-board RESET user button B2. If the error persists, verify that the SDA and SCL wires and pins were not mismatched.
2. Press the sequence "A", "A", "A" , "A" on the matrix keypad. This will set the current time to 00:00:00, the closing time to 00:00:00, and the empty and full distances equal to one another. Verify that the following text is displayed on the LCD:
    a. `"Space       Time"`
    b. `"N/0%    00:00:05"`
        i. The Time value should continue to increase at a rate of one tick per second.
    c. If the LCD remains at the expected output for Step 1, verify that the pin connections to the Matrix Keypad have all been properly established.
    d. If a new numeric value appears in the expected output for Step 1, verify that the rows and columns of the matrix keypad have not been swapped and that no wires have been crossed.
3. Press "#" on the matrix keypad.
    a. Verify that a "#" symbol appears on the LCD just to the left of the current time and that the Buzzer alarm produces a sound.
4. Press "#" on the matrix keypad.
    a. Verify that the "#" symbol disappears from the LCD and that the Buzzer no longer produces a sound.

b) Verifying Distance Sensor Behavior

5. Press "D" on the matrix keypad to return to the current time input state.
6. Press the sequence "C", "1", "2" on the matrix keypad to clear the previous time data and set the current time to 12:00:00.
    a. Verify that the following text is displayed on the LCD:
        i. `"Set current time"`
        ii. `"(24hr)  12:mm:ss"`

7. Press "A" on the matrix keypad to proceed to the next state.
8. Press the sequence "C", "1", "2", "0", "2" on the matrix keypad to clear the previous closing time data and set the closing time to 12:02:00.
    a. Verify that the following text is displayed on the LCD:
        i. `"Set closing time"`
        ii. `“(24hr)  12:05:ss"`
9. Press "A" on the matrix keypad to proceed to the next state.
10. Position the distance sensor so that it is aimed inside of the container whose distance is to be measured.
    a. Verify that the distance displayed in the bottom right corner of the LCD is a value between 2cm and 400cm that is approximately equal to the distance to the far end of the container.
11. Keeping the distance value on the LCD representative of the distance between the sensor and the far end of the container, press "A" on the matrix keypad to proceed to the next state.
12. Place an object just in front of the distance sensor so that the distance displayed just after "Set empty" on the LCD is less than 10 cm.
13. Keeping the distance value on the LCD representative of the distance between the sensor and the near end of the container, press "A" on the matrix keypad to proceed to the next state. Verify that text in the following format is displayed on the LCD:
    a. `"Space       Time"`
    b. `"nnn%   #12:01:20"`
        i. The Time value will not be an exact match and should continue to increase at a rate of one tick per second.
        ii. The value "nnn%" should be displayed as a number between "000%" and "100%".
        iii. The "#" symbol preceding the current time indicates that the alarm is armed.
14. Verify that moving the object within the container so that the percent value tends toward 0% when the object is farthest away from the sensor and toward 100% when moving closer to the sensor.
15. Wait for the displayed time to reach a time greater than the input closing time, keeping the object within the container.
16. Once the closing timestamp has been crossed, verify that the alarm is producing a sound and that the Space percent value is above 0%.
17. Remove the object from the container.
    a. Verify that the alarm is no longer producing a sound and that the Space value on the LCD is at 0%.
        i. If the Space value is at a low percentage such as 5%, the container or sensor may have shifted in place since the reference measurement was recorded. Increase the distance between the two slightly until the Space value reduces to 0%.
18. Press "#" on the matrix keypad.
    a. Verify that the "#" symbol on the LCD is no longer present.
19. Place the testing object inside of the container again.
    a. Verify that the Space percentage on the LCD is greater than 0%.
    b. Verify that the alarm is NOT producing a sound.
20. Keeping the object inside of the container, press "#" on the matrix keypad.
    a. Verify that the alarm is now producing a sound.
21. Press "#" on the matrix keypad to disable the alarm.

c) Verifying Watchdog Integrity

22. Press and hold any button on the matrix keypad for 30 seconds.
    a. Verify that the LCD is now on the "Set current time" screen.

10. Outcome of Implementation

The project implementation successfully met all design criteria.

The project implementation successfully completed all stages of the test plan.

11. Future Considerations

a) Identification of Shortfalls

The current design is missing some quality of life features for when the system needs to be rapidly configured or adjusted including manually inputting known distances and being able to go back a single configuration menu instead of restarting the entire process.

b) General Improvement

Further expansion of the system can be made to improve its resource efficiency and utility in minimizing food waste. These improvements include, but are not limited to the following:

- Keeping a record of factors that are expected to be constant between power cycles (closing time, full distance, empty distance) to use as defaults during the next power cycle.
- Recording capacity data for the container over the course of the day in order to identify when food demand is greatest. This data can be utilized to estimate a "target capacity utilization" metric to inform workers of when additional produce does or doesn't need to be added to the container.

12. Revision History Timeline

2021 November 19: Project Statement and preliminary Bill of Materials generated.

2021 November 20: Implemented the range testing program to control the new distance sensor peripheral in an isolated environment.

2021 November 22: Implemented a multithreaded testing program to verify the effective utilization of Mutexes to control access to shared resources by multiple threads.

2021 November 24: The distance sensor and LCD were implemented as two independent threads of the main program with Mutex-protected access to shared resources.

2021 November 25: The Matrix keypad code from Project 2 was merged into the main program of Project 3 as a third thread to allow for User Input to control the state machine elements of the system.

2021 November 26: The state pattern and signal to the buzzer output were implemented into the main program.

2021 November 29 – December 01: Comments were added to previously developed code.

2021 November 30: The first implementation of the watchdog timer was integrated into the main program.

2021 December 01: The watchdog timer was moved to the main method and modified to use the kick method due to the stop method not operating as intended.

2021 December 01: A test program to send a square wave signal to the Buzzer I/O pin was created.

2021 December 02: A duty cycle parameter was added to the buzzer test program and a sound effect was developed to play on the buzzer instead of a single constant note.

2021 December 02: The sound effect buzzer developed in the buzzer test program was merged into the main program utilizing two additional threads.

2021 December 03: Comments were added to the newly developed buzzer code.

Appendix A: References

SparkFun Electronics, "Ultrasonic Ranging Module HC - SR04," *HC-SR04 Datasheet by SparkFun Electronics*. [Online]. Available: https://www.digikey.com/htmldatasheets/production/1979760/0/0/1/hc-sr04.html. [Accessed: 05-Dec-2021].