CSE321_Project3 Security System

Table of Contents

<u>CSE 321 Project 3</u>
<u>Security System</u>

<u>By:</u> Patryk Kasza

Person #: 50275611

Email: pskasza@buffalo.edu

<u>Introduction:</u>

This project was designed in order to improve the overall safety of a household, or any building for that matter. It uses an ultrasonic sensor to detect movement with an area. The device can be set to locked or unlocked through the onboard button, which will determine whether or not the audio module will buzz if movement is detected. This allows the user to know that something, or someone is nearby only when they want to know. For ease of handling, an LCD display is incorporated to display whether or not the device is locked, as well as the corresponding distance that the ultrasonic sensor picks up. The onboard button can be used to lock and unlock the device, allowing you to use it when you know there will be movement in the area, and you don't want to be alerted.

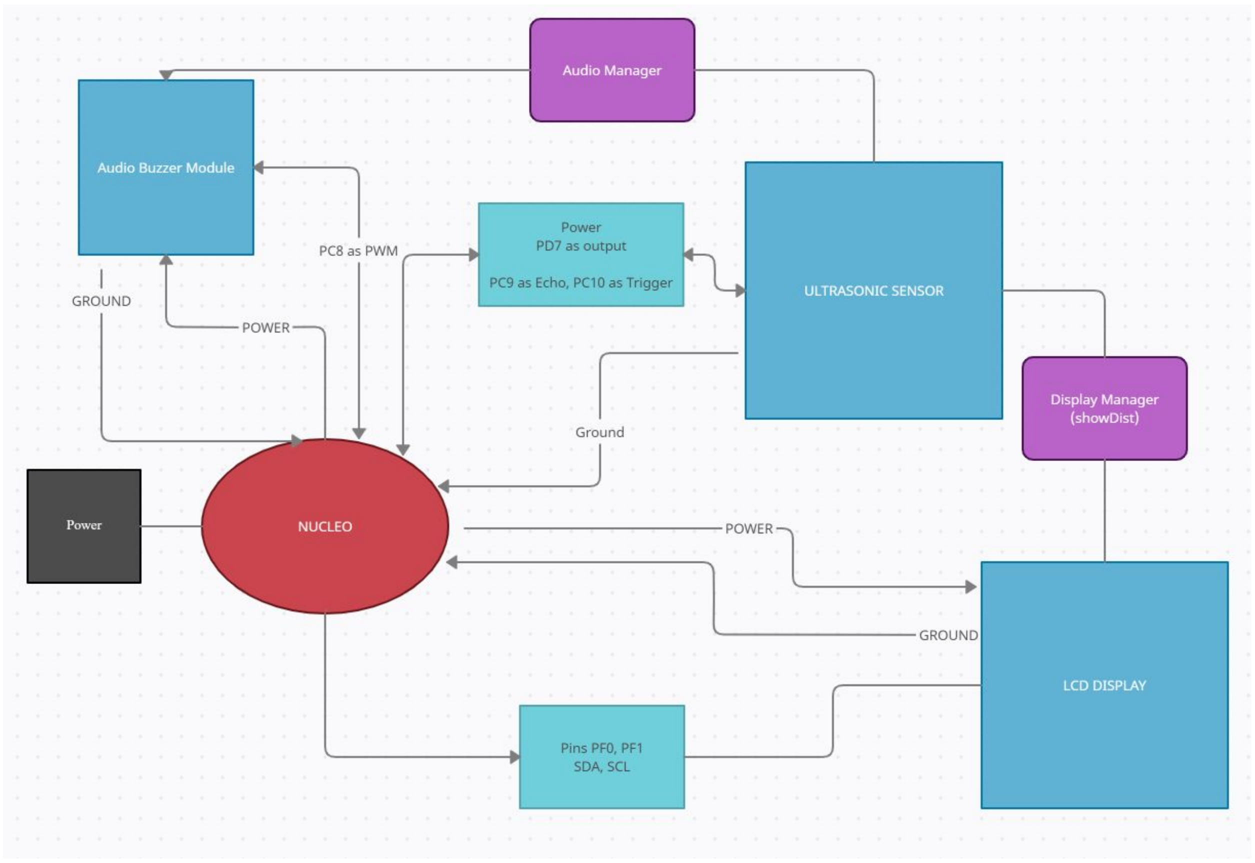
Project Requirements

1. **Inputs:**
    a. Onboard user button
    b. Ultrasonic Sensor
2. **Outputs:**
    a. Audio Module
    b. LCD display
3. **Functions:**
    a. ISR, buttonLock
    b. AudioHandlerOFF
    c. AudioHandlerON
    d. ShowDist

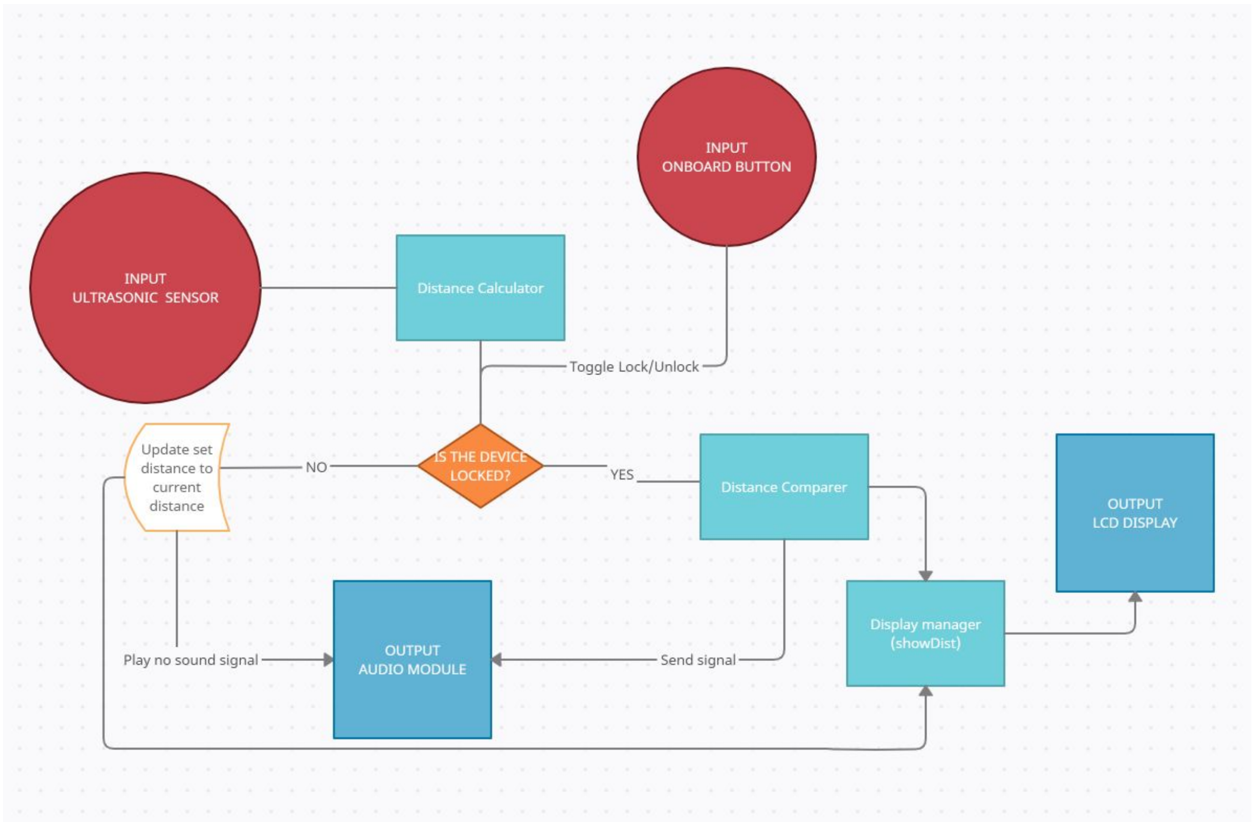
4. **Constraints:**
    a. No abnormal temperature in the environment to prevent ultrasonic sensor malfunctions
    b. On startup no movement allowed of objects in the sensors field of view
    c. No movement of the ultrasonic sensor
    d. Must run forever
    e. Sensor max distance, 400cm. Minimum distance 2cm.
5. **Specifications:**
    a. Buzzer only buzzes when movement is detected from the ultrasonic sensor
    b. LCD queue on activation of the ultrasonic sensor and deactivation(When the sensor goes on and off, different visuals provided)
    c. Movement detected only in the field of view of the ultrasonic sensor
    d. On detection visual and audible queue simultaneously provided

Design Process Review:

Block Diagram:



Flow Chart

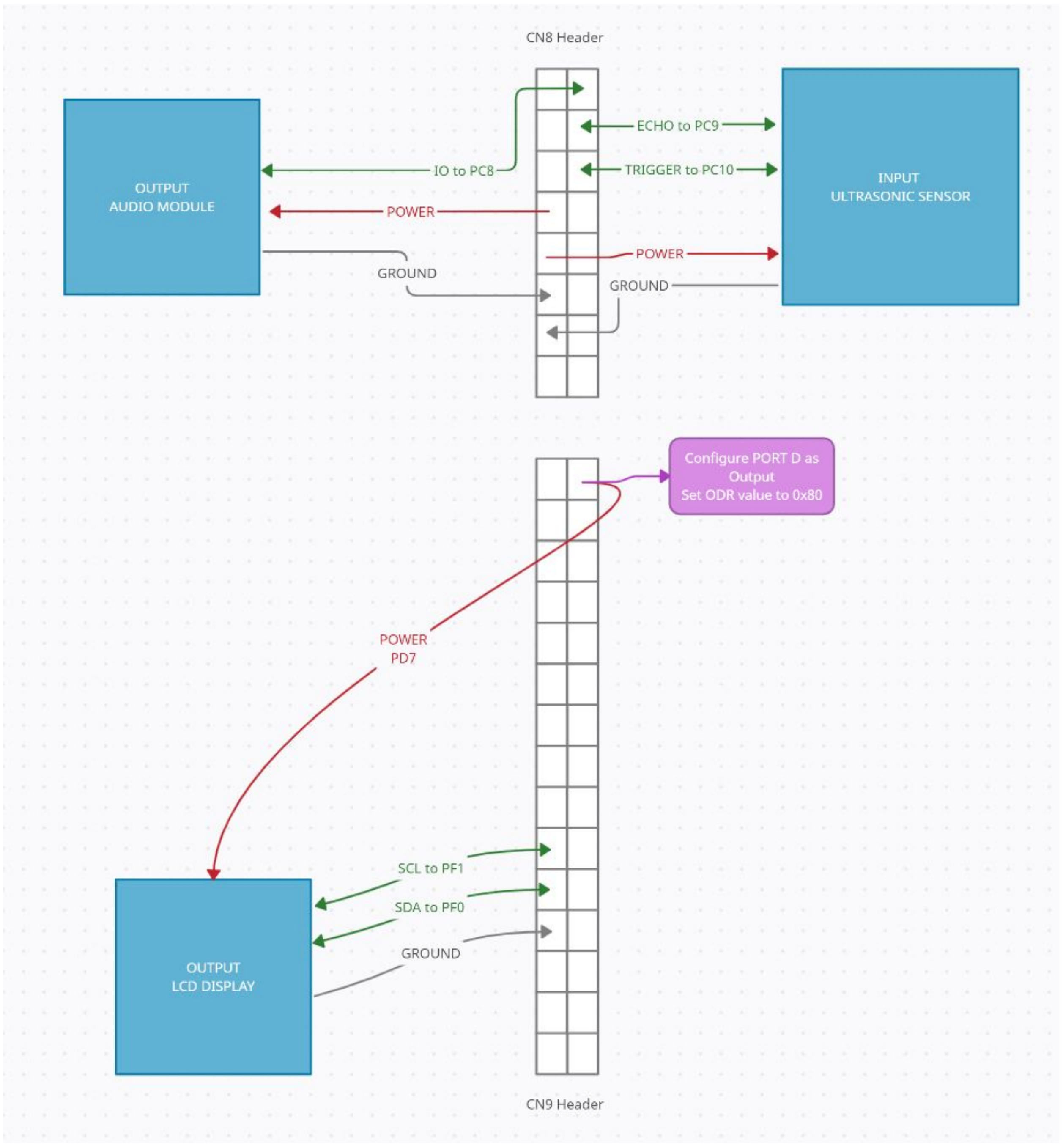

INPUT ONBOARD BUTTON
INPUT ULTRASONIC SENSOR
Distance Calculator
Toggle Lock/Unlock
Update set distance to current distance
NO
IS THE DEVICE LOCKED?
YES
Distance Comparer
OUTPUT LCD DISPLAY
Play no sound signal
OUTPUT AUDIO MODULE
Send signal
Display manager (showDist)

BOM

1. NUCLEO-L4R5ZI: Microcontroller, main component
2. Micro USB cable: Connect/power microcontroller
3. SainSmart HC-SR04 Ranging Detector Mod Distance Sensor: To detect movement in the area
    a. https://www.amazon.com/gp/product/B004U8TOE6/ref=ppx_yo_dt_b_asin_title_o00_s01?ie=UTF8&psc=1
4. ARCELI 5pcs DC 3.3-5V Passive Low Level Trigger Buzzer Alarm Sound Module: To give audio queue.
    a. https://www.amazon.com/gp/product/B07MPYWVGD/ref=ppx_yo_dt_b_asin_title_o00_s00?ie=UTF8&psc=1
5. At least 11 male to female jumper wires: To connect external peripherals
6. LCD 1802 model: To give a visual display

Instructions

Schematic



Setup instructions

Button setup: Associate it with an ISR. Give it a thread and an eventqueue. Initialize this in main. Follow the Test Plan below to see if it works properly.

Audio Module: Set up the IO pin as PWMOUT. You will use (module object name).write(0) or .write(1.0f) to trigger noise signals sent to it. Create a function for each respective mode.Follow the Test Plan below to see if it works properly.

Ultrasonic Sensor: First configure the pins to trigger and echo as described in the schematic. Create a timer object as well to measure time between signal sent and received. Then give a high signal to trigger pin for 10us. Afterwards wait for echo to send a signal(echo high). Start the time and wait for echo low. Stop the timer and convert the time to cm, or your measurement of choice. Follow the Test Plan below to see if it works properly.

LCD Display: Import the library, 1802.h. Initialize the CSE321_LCD as described in the library. Start the LCD in main. Create global strings to later append to. Try printing the global strings. Follow the Test Plan below to see if it works properly.

While not set to beep, you want to constantly update the global variable for distance. Once it is set to beep, you can compare with the latest updated distance.

Now decide the minimum distance for it to beep. Default is 5 cm.

How to use:

You can use this device as a basic movement detection system. You can lock it, and it will give you a high pitched audio queue when movement is detected. Can be used when you're sleeping alone in your private space, it will surely alert you if someone crept in. Can also be used outside, but it isn't recommended as the max distance is only 400cm. Know what's happening around your area, and feel a little more safe!

Test Plan

Testing onBoard Button:

1. Configured properly?
    a. Setup ISR, printf(Hello) when triggered

Testing buzzer:

1. Setup correctly?
    a. (buzzer name).write(0) and .write(1.0f), alternate between making noise and not making noise.
2. Put in logic for onboard button to alternate
    a. Does the buzzer alternate between making noise and not?

Testing UltraSonic Sensor:

1. Setup correctly?
    a. Print the distance you calculate
    b. Use common intuition to see if it reads correctly
        i. Put hand in front of sensor
            1. Shorter distance recorded?
        ii. Measure distance from sensor to facing object
            1. Does it match?

Testing LCD:

1. Is the LCD displaying correctly?
    a. lcd.print(hello), lcd displays hello?
2. Print a global string variable
    a. Does lcd display correctly?
3. Update that global string variable with input
    a. Does lcd display correctly clear and update?
4. Associate offsets with strings representing locked/unlocked
    a. Are the offsets correct?
    b. Does it update with the distance continuously?

Testing watchDog:

1. Start timer, don't ever kick.
    a. Does it restart?
2. Setup conditions for kicking the watchdog, max distance is 400cm. Continue the timer on those conditions.
    a. Printf everytime its kicked
    b. Does it kick under correct conditions?

Revision History

Date	Revision	Changes
31-Oct-2020	Initial Definition of Problem/System	Initial Start
1-Nov-2020	Stage 2 Close Out	Identified preliminary constraints and specifications. Created Repo
8-Nov-2020	Stage 3 Close Out	Plan laid out for project. BOM nearly completed
12-Nov-2020	Stage 4 Close Out	Basic testing for onboard peripherals done.
22-Nov-2020	Stage 5 Close Out	Audio buzzer tested and configured. Failed attempts at configuring ultrasonic sensor.
10-Dec-2020	Final Submission Deadline	Ultrasonic sensor configured. LCD display configured. Logic all added to handle appropriate inputs from the ultrasonic sensor.

Future Design Considerations.

Shortfalls of the System:

This system can’t track movement in outdoor spaces as I had anticipated. The range is surprisingly short, at around 400cm. Also it can’t function well for moving object detection as the sensor is sensitive to moving it. It could work, just not to the capability I had imagined. That being said, it can only provide movement detection indoors, in a stationary position. With such a short range it will be hard to implement this design in many scenarios.

Communication Feature:

If I could implement the CAN feature, in addition to adding more sensors I could create a system where each sensor “talks" to each other. Providing more range and field of view for the system. It would consist of them sharing when one detects movement, so none of the others override the audio buzzers input.

Memory Management:

With only one ultrasonic sensor, there was no need to give its own separate thread. So the memory management is simplistic and good for the scope of my prototype.

However I could synchronize all my global variables with appropriate mutexes. That way it will be easily scalable with an increased amount of sensors, or threads for other uses.

Direct Memory Access:

I could definitely implement this to update global variables, the distance in particular. It will increase the speed of updating this value as well as make it much safer to manage across multiple threads.

Potential Design Improvements:

In terms of cost efficiency, including a motor to rotate the sensor would be much better than the cost of more sensors. It would require extra logic and initial start up configuration. However including this would help cover a wider field of view and improve the systems general security capabilities.

References

Ultrasonic config:
https://os.mbed.com/components/HC-SR04/
http://www.emcu.eu/understand-the-way-to-use-hc-srf04-on-stm32-nucleo-board-and-mbed/

Ultrasonic Datasheet:
https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf