**Social Distance Encouragement System**

**Report Created by:** Amanda La

**CSE321 Fall 2020**

**Lab Due:** December 10, 2020, 11:59 pm

**Lab Closes:** December 11, 2020, 11:59pm

**Table of Contents**

*Cover page ....................................................................................................................1*

*Introduction.....................................................................................................................2*

*Project Requirements.....................................................................................................3*

*Features...............................................................................................................3*

*Specifications.......................................................................................................3*

*Solution Development.....................................................................................................3*

*Block Diagram......................................................................................................4*

*System Diagram...................................................................................................5*

*User Instructions.............................................................................................................5*

*Bill Of Materials..................................................................................................5 -6*

*Schematic Diagram............................................................................................6-7*

*Set-up Instructions................................................................................................7*

*Guide to use the system.......................................................................................8*

*Test Plan.........................................................................................................8 - 10*

*Revision history..............................................................................................................10*

*Future Design Considerations........................................................................................10*

*Shortfalls for System Design................................................................................10*

*Communication Feature.......................................................................................10*

*Memory Management..........................................................................................11*

*Direct Memory Access.........................................................................................11*

*Improvement Recommendations.........................................................................11*

**Introduction**

*This project serves to tackle the challenge of encouraging social distancing without having to communicate to the people around the system that they are within six-feet of another person. Social distance is encouraged through this system by producing a buzzing noise when an object is detected within the range of the IR sensor. The buzzing noise will not stop until the object has moved out of the range of the IR sensor. This system serves as a not-so-subtle way to encourage and remind those around the system to practice social distancing.*

**Project Requirements**

*This system utilizes an IR sensor, an LED, and a buzzer. Watchdog is incorporated into the system to reset the system into a safe state. Synchronization is integrated through the use of ISRs and an event queue. The ports B and C were set up bitwise where port B is used as an output and port C is used for inputs.*

**Features**

*Features of this design include: a button capability to allow the user to confirm whether the integrated buzzer is functional at any point of the program, a method of informing the user whether the IR sensor has recognized an object within its’ range as well as whether an object is not detected within that same range.*

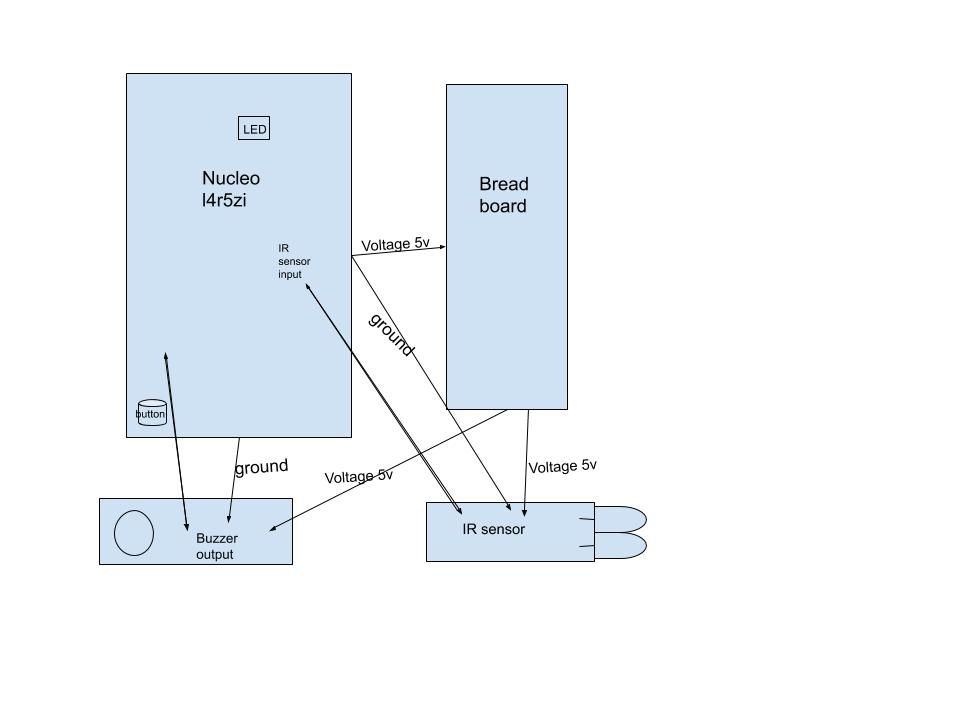
**Specifications**

*The system contains an IR sensor and a user-onboard button which act as inputs as well as a buzzer and an onboard LED acting as outputs. The systems’ goal is to detects an object within the systems’ range and act according to the presence of an object. If an object is visible to the sensor, then the system will trigger the buzzer to produce recognizable noise. If an object is not detected within the IR sensors’ range, then the system will disable the buzzers’ noise. The noise produced by the buzzer should continue for as long as the object exists within the range of the IR sensor. Whenever an IR sensor input is received, the system will turn on a buzzer allowing the user to acknowledge that the IR sensor input was received by the system. The onboard LED acts as a sanity check for our user-onboard button input and buzzer output. When the onboard button is pressed, the onboard LED will turn on and the buzzer will also turn on to produce noise. When the button is pressed in an effort to turn the light and buzzer off, it will produce that effect.*

**Solution Development**

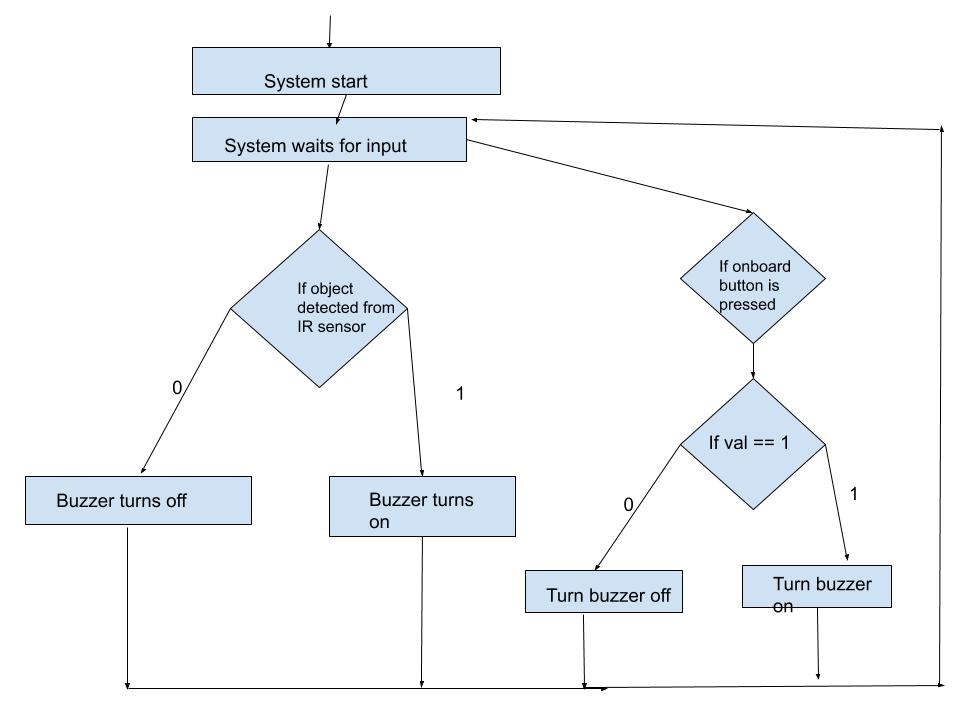
*Creating this project required utilizing key elements such as ISR functions and an event queue to prevent mutex errors from occurring as well as implementations of test functions to ensure hardware was functional. Another key element included utilizing testing practices such as changing the pin for a hardware piece in the event of a failed interaction between software and hardware as well as utilizing print statements to ensure inputs and or outputs were registered.*

**Block Diagram**

****

**System Diagram**

*ASM*

****

**User Instructions**

**Bill of Materials**

*Hardware components*

*Six (6) male to female jumper cables*

*One (1) male to male jumper cable*

*One (1) USB 3.0 to a Micro USB cable*

*One (1) IR Sensor*

*One (1) 3-pin Buzzer*

*One (1) Solderless breadboard*

*Software components/libraries*

*These libraries come with mbed, they were automatically added to my file.*

*InterruptIn.h*

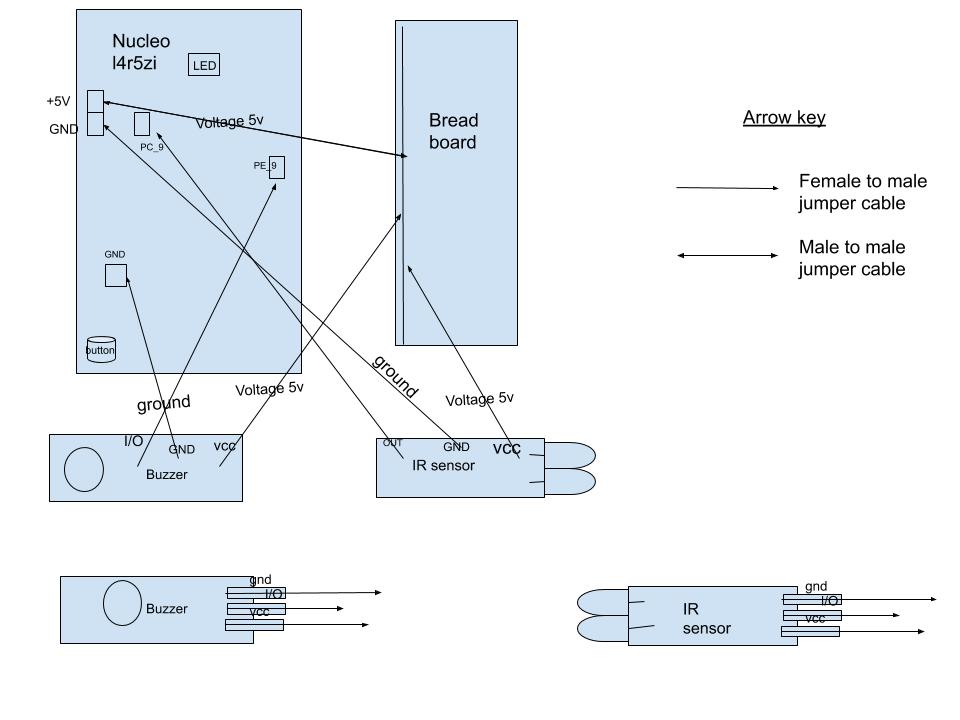
*PinNamesTypes.h*

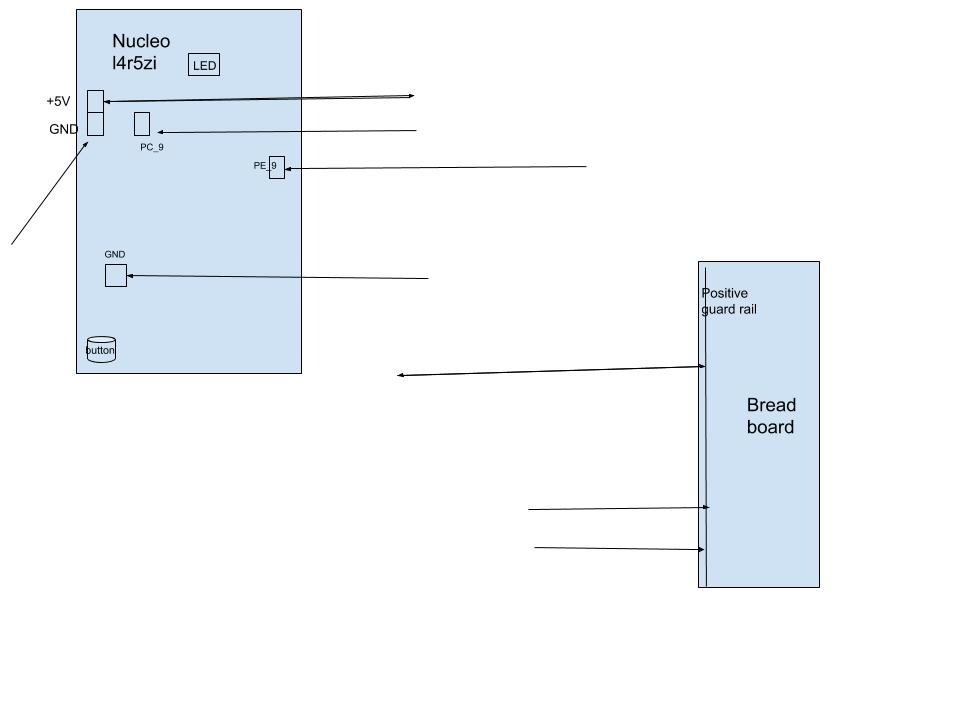
*mbed.h*

*mbed\_thread.h*

*mbed\_wait\_api.h*

**Schematic**

**

**

**Setup Instructions**

*Firstly, grab a male-to-male jumper cable and locate the +5V pin on the Nucleo-L4R5ZI. Once located, plug any end of the male-to-male jumper cable into the +5V pin and the other end into the positive rail of the breadboard. Then, grab three male-to-female jumper cables and attach the female ends to the IR sensors’ GND, OUT, and VCC. Do the same for the buzzer with the left over male-to-female cables. Then, plug the male end for the cables’ attached to the VCC of both devices into the positive guard rail. Locate two GND pin locations on the Nucleo-L4R5ZI. Connect the male end of cables’ attached to the GND of both devices into the located pins. Finally, plug the male end associated with the OUT port for the IR sensor into pin PC9 on the Nucleo-L4R5ZI and the male end associated with the I/O port for the buzzer into pin PE\_9. All jumper cables and ends of jumper cables should be attached to something by now. Plug the USB into computer and into top of Nucleo-L4R5ZI and system should be fully constructed.*

**Guide to use the system**

*Before building your program, ensure that only one light is on for your IR sensor. If two lights are on, you will need to physically recalibrate your sensor or change the direction your sensor is pointing to so that it is not pointing to an object it can detect (this is based on the range of the sensor you have). Once your IR sensor only has one light on, test the distance that your IR sensor can detect an object by placing either a notebook or your hand in front of the sensor and slowly move towards it until the second light pops up and away from it until the second light turns off (object detected when the second light turns on). If you are 1cm away from the sensor and a second light is not turning on, try to obtain a second IR sensor and replace this piece since your sensor might not be functional. Once you have found a testable distance range, we can start using the system.*

*Build the program and wait for serial monitor to say “------start-----“. A buzzing noise should start and stop indicating system is ready for use at the same time. The buzzer should be off by now. Once these two events happen, we can begin testing the system. Press the bottom left user button on the Nucleo-L4R5ZI to turn on the buzzer and ensure the buzzer is functional. Press the same button again to turn off the buzzer.*

*Now we can test our systems’ goal. Your IR sensor should still have only one light on. Since the IR sensor provided has a limited range, trigger the system by physically moving your hand within the range established previously in front of the sensor. If your sensors’ second light turns on, the buzzer should turn on as well and the serial monitor should print that an object as been detected. Hold your hand in front of the IR sensor where it is able to successfully detect your hand. As you hold your hand, the buzzer should continue to buzz. Move your hand out of range of the sensor, and the buzzer should stop buzzing and inform through the serial monitor that an object is no longer detected.*

**Test Plan**

*Software test plan – IR sensor*

*When the IR device detects an object within its’ range, the interrupt ISR function will output “object detected” to the serial monitor, informing the user that an object has been detected by the IR sensor. This test helps ensure that our IR device is properly sensing objects and informing the software that an object is detected in real time.*

*Hardware test plan - Buzzer*

*When the user onboard button corresponding to PC\_13 is pushed, the buzzer will turn on to confirm that our buzzer is working properly with an input ISR triggering its’ use.*

*Hardware/Software test plan – IR Sensor & Buzzer*

*When the IR sensor detects an object within range of the IR sensor, the buzzer will turn on indicating an object exists within the range of the IR sensor. When an object does not exist within the range of the IR sensor, the buzzer should make no noise. This ensures our buzzer and IR sensor are functional and are communicating through the software.*

Analysis Plan Tests

1. *Hardware buzzer test*
   1. *The buzzer will be tested*
   2. *The buzzer is being tested to ensure that the hardware received is functioning properly independent of the system*
   3. *I will test the buzzer by developing a function which will trigger the buzzer, setting it off*
2. *Software IR test*
   1. *The IR device will be tested*
   2. *The IR is tested to ensure the hardware received is functioning properly independent of the system and is being read on the software side*
   3. *I will test the IR device by pointed the IR at different objects, and outputting to the console the temperature of the object. This test will be consolidated with a physical thermometer. Both measurements will be compared through conversions.*
3. *Software buzzer test*
   1. *The buzzer will be tested*
   2. *The buzzer is tested to ensure that it is being read in the software side*
   3. *I will test the buzzer by printing to the console after each function call to the buzzer to ensure that the function executed and terminated from the call.*
4. *System test*
   1. *The system will be tested*
   2. *The system is tested to ensure that each piece is working in tandem and producing the desired output*
   3. *I will test the system by measuring six feet and pointing the system towards me. I will stand six feet away and if the system does not activate the buzzer, then this test passes.*
5. *System test* 
   1. *The system itself is being tested*
   2. *The system is being tested to ensure it is functioning the way I want it to*
   3. *The system will be tested by pointing the IR device at myself, and measuring a six foot distance from me to the device. If I approach within six feet, then the buzzer should go off.*

**Revision History**

|  |  |  |
| --- | --- | --- |
| **Date** | **Revision** | **Changes** |
| 31-Oct-2020 | Initial Definition of Problem/System | Initial Start |
| 8-Nov-2020 | Stage 2 Close Out | *Purpose, ASKs, and BOM established* |
| *15-Nov-2020* | Stage 3 Close Out | *ASM developed, possible change of materials* |
| *22-Nov-2020* | Stage 4 Close Out | *Materials changed, tests developed for hardware and software* |
| *30-Nov-2020* | Stage 5 Close Out | *Coding for project in testing, hardware functioning with software, ISR Queue overflow error* |
| 10-Dec-2020 | Final Submission Deadline | *Coding for project finalized, system functional* |

**Future Design Considerations**

**Shortfalls for System Design**

*The application does not meet the criteria of measuring at a six-foot distance due to hardware obtained. The application does however serve as a scaled version which can be scaled up to six feet depending on the hardware obtained. The application also only serves as a ‘measuring stick’ for whichever direction the IR sensor is pointing in, therefore not capturing all directions around the user.*

**Communication Feature**

*This interface can integrate a SPI communication feature because the system is mainly concerned with the input information from the IR sensor. This helps save on cost and improves efficiency of the system by focusing on the information that needs to be handled.*

**Memory Management**

*I made the choice of utilizing an Event Queue to manage the memory of the interrupt threads. I also established how much space would be needed for the declared EventQueue so that the system allocated just as much memory as would be needed. One memory management control tool that could have been incorporated into the system is to utilize calloc with the system. Calloc would better the system by allowing the system to allocate and initialize memory for the EventQueue rather than just allocating memory as we currently are doing. This would help prevent any issues regarding mismanagement of memory and clarify which areas of the program we should be concerned with.*

**Direct Memory Access**

*A DMA integrated with our IR sensor would improve our system by making our response time to receiving the input from the IR sensor faster, helping our program function better in real time.*

**Improvement Recommendations**

*A feature that could be improved within this system is how the system currently addresses bounce. Improving how the bounce is addressed can help avoid ISR Queue overflow errors within the system which results in the system relying on watchdog to stabilize the state of the system. By improving bounce addressment, the system can run with less possible errors.*

## 