PROJECT 3 (SILENT STUDY NOTIFICATION SYSTEM)

Motaz Khalifa

UNIVERSITY AT BUFFALO [CSE 321 | FALL 2020]

Table of Contents

Introduction 2
Project Requirements 3
  Specifications 3
    Inputs 3
    Outputs 3
    Functions 3
  Features 3
Solution Development 4
  Block Diagram 4
  Functionality Diagrams 5
User Instructions 6
  Bill of Materials 6
  Schematic Diagram 7
  Instructions to build 8
  Instructions to use 9
  Test Plan 10
Development Timeline 11
Future Considerations 12
  Identification of shortfalls 12
  Communication Feature 12
  Memory Management 12
  DMA 12
  General Improvement 12
References 13

Introduction

This application is a system that notifies the user if he is being too loud inside a study place (ex: library) for educational purposes. It is designed such that if a certain sound threshold is broken the system will start a silent alert that notifies the user that he should remain quiet. The alert consists of an LED that lights up, an LCD screen that asks the user to remain quiet, and a vibration motor that starts working. The alert could be silenced by pressing a push button.

This tool could be distributed across tables inside a library so that each person could be notified if he is being too loud.

Project Requirements

The project requires utilization of a sound detector and a vibration motor to be able to detect that the environment surrounding it is too loud, and send a signal to the user.

Specifications

Inputs

- Sound above a certain level

Outputs

- Notifies user that he is being too loud

Functions

- Registers audio loudness above a certain threshold that turns on an alert
- Lights up an LED when it is loud
- Prints to an LCD when it is loud
- Starts the vibration motor when it is loud

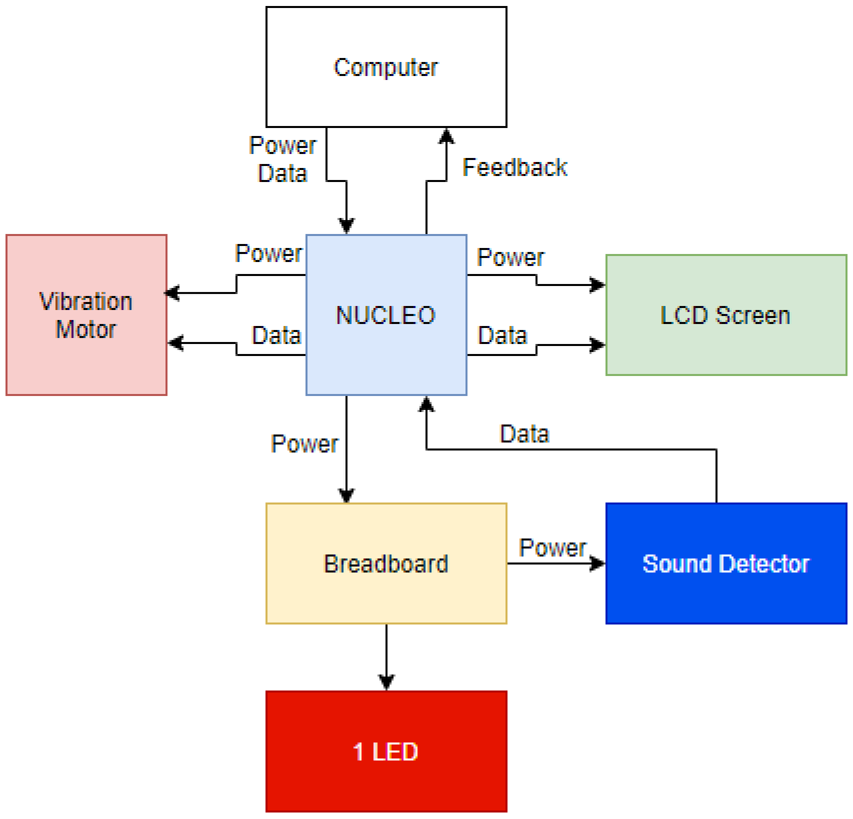
Features

- Sound detector for measuring loudness
- LCD shows current state of system (Loud or Quiet)
- LED lights up when it is loud
- Vibration motor

Solution Development

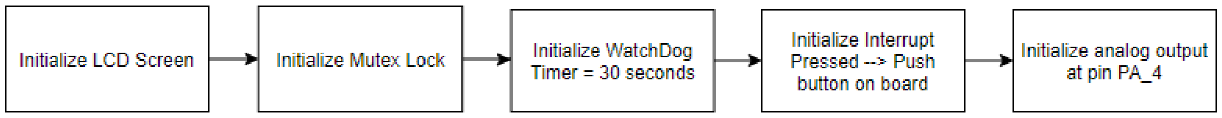
In this project there are different approaches to reading inputs from devices and writing outputs. Bitwise integration was used to setup the LED and configuration of sound detector, while mbedOS API Analogout was used to power up the vibration motor. Synchronization was used when necessary to avoid conflicts.

Block Diagram

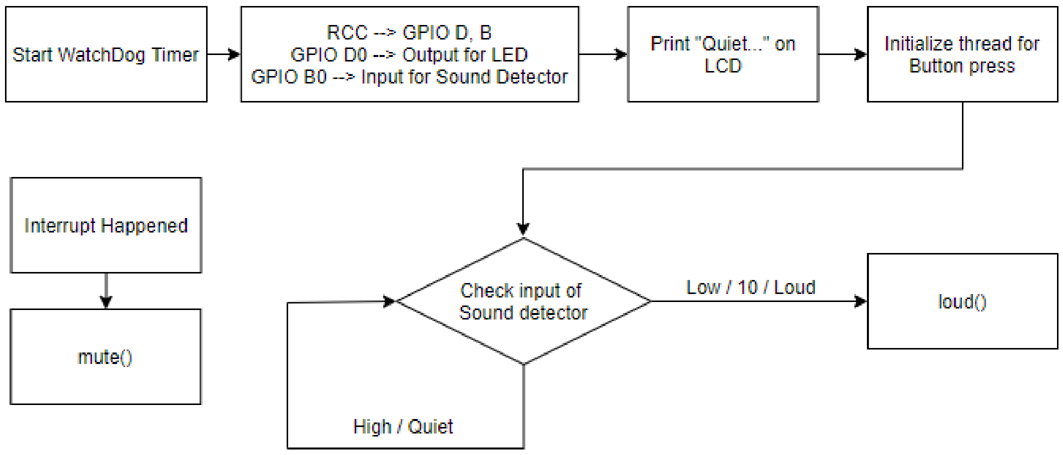


Functionality Diagrams

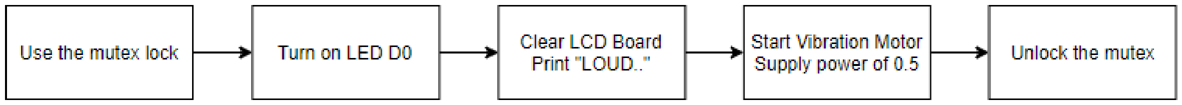
Initializations


Initialize LCD Screen
Initialize Mutex Lock
Initialize WatchDog Timer = 30 seconds
Initialize Interrupt Pressed --> Push button on board
Initialize analog output at pin PA_4

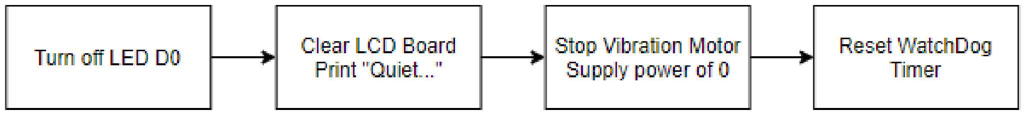
Main


Start WatchDog Timer
RCC --> GPIO D, B
GPIO D0 --> Output for LED
GPIO B0 --> Input for Sound Detector
Print "Quiet..." on LCD
Initialize thread for Button press
Interrupt Happened
mute()
Check input of Sound detector
Low / 10 / Loud
loud()
High / Quiet

Loud


Use the mutex lock
Turn on LED D0
Clear LCD Board Print "LOUD.."
Start Vibration Motor Supply power of 0.5
Unlock the mutex

Mute

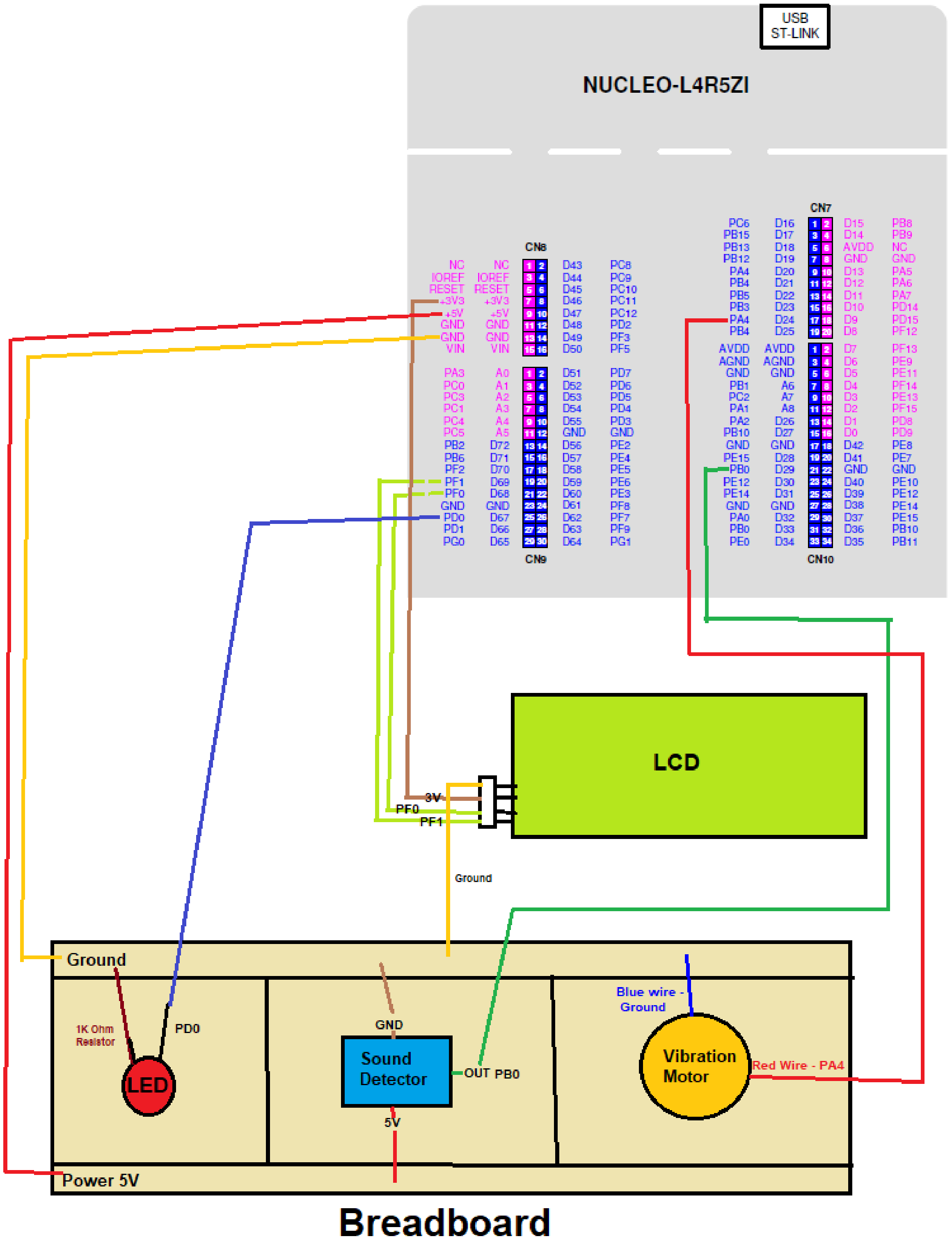

Turn off LED D0
Clear LCD Board Print "Quiet..."
Stop Vibration Motor Supply power of 0
Reset WatchDog Timer

User Instructions

Bill of Materials

- Nucleo-L4R5ZI
- USB A to Micro USB B cable
- 16x2 1802 LCD
- 1802.h
- 1802.cpp
- Breadboard
- Jumpers/Wires
- 1 Red LED
- 1 Resistor (1kOhm)
- 1 LM393 Sound Detection Sensor
- 1 DZS Elec Mini Vibration Motor
- Small screwdriver or pin to adjust the knob on the sound detector

Schematic Diagram



Instructions to build

a) Breadboard
   1. Connect the upper rack with the GND on the Nucleo-Board
   2. Connect the lower rack with the 5V+ PWR on the Nucleo-Board

b) LED
   1. Place the LED on the breadboard
   2. Connect the shorter side with a 1K ohm resistor
   3. Connect the resistor to the GND side of the breadboard
   4. Connect the longer side to pin PD0 on the Nucleo-Board

c) LCD Screen
   1. Connect the GND pin to the GND side on the breadboard
   2. Connect the VCC pin to the 3V pin on the Nucleo-Board
   3. Connect the SDA pin to pin PF0 on the Nucleo-Board
   4. Connect the SCL pin to pin PF1 on the Nucleo-Board

d) Vibration Motor
   1. Connect the red wire to pin PA4 on the Nucleo-Board
   2. Connect the ground wire to GND side on the breadboard

e) Sound Detector
   1. Connect the GND pin to the GND side on the breadboard
   2. Connect the +5V pin to the 5V+ PWR side on the breadboard
   3. Connect the OUT pin PB0 on the Nucleo-Board

Instructions to use

1. After connecting everything together, when running the program for the first time, user will first have to calibrate the sound detection sensor before using it. First run the program, then use a small screwdriver to **slowly** rotate the pot on the sound detector clockwise until the LED lights integrated on the sound detector up, then slowly rotate it again counter-clockwise until it just turns off. This might send the program into the alert mode unnecessarily, so re-run the program again from mbed studio after finishing this step.
2. When the application is running and the surrounding area is quiet, the LCD screen should say “Quiet...”, and the LED on the breadboard should be shut off, same as the vibration motor.
3. To send the application into the loud mode, try saying something loud or clapping your hands near the sensor, this should be sufficient to send it into the loud mode.
4. While in the loud mode, the LCD screen will print “LOUD..”, the LED will light up, and the vibration motor will start working.
5. To mute the application, press on the blue push button on the Nucleo-board.
6. This will reset the application and send it into the quiet mode again.
7. For further testing, you **do not** have to re-calibrate the sound detector again.
8. If 30 seconds passed without clicking on the push button, the watchdog timer will reset the program.

Test Plan

a) Before Installing System
   - Test the LED individually to make sure it is working.
   - Test the push button individually to make sure both the interrupt and the button are working.
   - Test the vibration motor individually by supplying power to it using AnalogOut.
   - Test the sound detector individually by printing to mbed output when loud.
   - Test the LCD screen individually by printing numbers to it.

b) After Installation
   - Clap hands near the sound detector (input)
     i. LCD should print “LOUD..”. (output 1)
     ii. LED should light up. (output 2)
     iii. Vibration motor should start working. (output 3
   - When sound detector is in the loud mode, click on the push button on the Nucleo-board, this should mute the alert system.
   - The push button is the only part of the application able to reset the watchdog timer
     i. Run the program, and wait for 30 seconds without making any sounds. The result should be that the program will print the “Start” label again and reset all output devices to the quiet mode.
     ii. Run the program, clap hands near the sound detector, and wait for less than 30 seconds without pressing the push button, the watchdog timer should again reset the program.
     iii. Run the program, wait for 25 seconds and press the push button (irrespective to whether it is in the loud or quiet mode) and press the push button, this should reset the watchdog timer.

Development Timeline

Date	Revision	Changes
31-Oct-2020	Initial Definition of Problem/System	No partner was selected
2-Nov-2020	Stage 2 Close Out	-Created repository -Made initial system idea, a system that notifies a person if he is being too loud inside a library. -Made a list of system requirements and needed tools -Made a list of initial constraints
9-Nov-2020	Stage 3 Close Out	Didn't submit stage, ordered parts necessary for the project. -Sound Detector -Vibration Motor - Already had LCD and LED from project 2
16-Nov-2020	Stage 4 Close Out	-Committed skeleton files for project to git -Received parts
23-Nov-2020	Stage 5 Close Out	Didn't submit stage, started doing research on the new parts
11-Dec-2020	Final Submission Deadline	Completed project

Future Considerations

Identification of shortfalls

The initial design of the project was that the intensity of the vibration motor's response should vary correspondingly with how loud the surrounding environment was. But the device turned out to be a ticker that only sends either a 1 or 0, cancelling the possibility of knowing the intensity of sound registered.

Communication Feature

CAN could be used to connect the vibration motor with the sound detector to turn on the motor when any sound is detected.

Memory Management

The project does not consume much memory, the EventQueue handles the interrupt in a way that does not consume much memory.

DMA

There are currently no chances in sight to use DMA in a way that might help the project.

General Improvement

A new feature could be added in the future, and that is adding more sound detectors but with different threshold values. This way the system would be able to respond back with intensity corresponding to how loud the environment is.

References

- https://drive.google.com/file/d/1N3nr2m25jU2xqbqBTnGvhL9j5vlGCO2N/view
- https://randomnerdtutorials.com/guide-for-microphone-sound-sensor-with-arduino/
- https://www.electroniclinic.com/arduino-micro-vibration-motor-arduino-vibration-motor-code-interfacing/