Zachary Wolf

CSE321

Project 3

Fall semester, 2021

Purpose: To create an embedded system that provides feedback about nearby obstacles to the visually impaired.

Table of Contents

Introduction

Embedded systems have a very large amount of potential uses for a wide variety of applications. It's not an exaggeration to say that devices built with them can have a noticeably positive impact on the world. To this end, the following system was constructed as a way of providing feedback to individuals who are visually impaired.

Overview of specifications and features

The device to be built is designed to provide accessibility support, specifically, to the visually imparied. The device will use an infrared sensor in order to detect nearby objects and will then provide feedback to the user via vibration motor. The system will also include green and red LEDs with the red being on when not near an object, and green activating when it is. These can be used to check if there is an issue with the vibration motor and for people who have poor eyesight, but not no eyesight. For someone who has poor eyesight, this device can act as additional feedback for navigating rooms. The device could be used to detect an object and will alert the user using the vibration motor which should provide a distinct signal to the user, while being relatively non-distracting to people nearby.

Use of the watchdog and how it was incorporated

The watchdog is initialized in the main function upon running the program on the device. It will revert the system to the state that it is in if there are no inputs within 5 seconds. Receiving a signal, or losing the signal from the infrared sensor and their associated interrupts will both 'feed' the watchdog, resetting its timer. The use of this implementation, is that if the board were to receive a signal to activate the 'object detected' function, but somehow missed the interrupt caused by the cessation of signal from the infrared sensor, the system would not be stuck in a state with the vibration motor on.

Synchronization techniques and how it was incorporated

Within the system, it is crucial that the thread that would have managed the infrared sensor's inputs doesn't update its status of detected or not detected while the thread that handles the bitwise drivers is in the process of updating which ports are turned on or off. This would have been handled by implementing a lock that forces the system to wait for the bitwise drivers to finish updating before changing them again. However MBED refused to acknowledge any thread declarations as valid resulting in these threads not being implemented. This made synchronization impossible as there was not more than one thread to synchronize between, and the implementation of locks nonsensical.

Bitwise driver control and how it was incorporated

Bitwise drivers were used for the controlling of two external LEDs. These LEDs provided redundancy in the form of indicating whether or not an object was detected by the infrared sensor. They also acted as an additional feedback system for if an object was close or not, which could be useful to a visually impaired person if they still have some sight.

Critical section protection and how it was incorporated

In the design, if the object detected and object not detected functions were to happen simultaneously, perhaps as a result of the infrared sensor detecting and then not detecting something in rapid succession, the bitwise control could set the LEDs to a state that they are not supposed to be with, either with both off or both on.

The implementation's use of interrupts helped to avoid this, as both the detected and not detected code segments were made into their own helper functions, and were only triggered by interrupts. Due to the nature of interrupts, the code executing could not resume until an interrupt has finished, thus the system would not perform the bitwise activations simultaneously.

Thread/tasks and how it was incorporated

The original design was for two threads to exist in the system. One for the inputs from the infrared sensor, and the other to manage the LED and vibration motor outputs. However, this idea was made undoable when MBED studio refused to acknowledge ‘Thread’ as valid. As a result, threads could not be declared and thus could not be used. A considerable amount of time was spent attempting to debug the issue with no success. Thus the design was adapted to function without it.

Interrupts and how it was incorporated

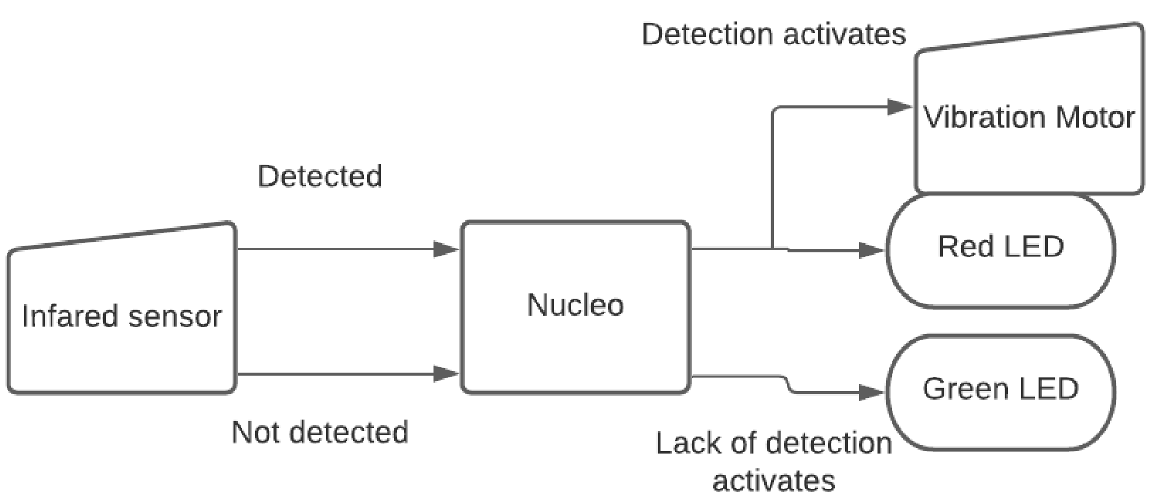
Interrupts were used to process the signals from the infrared sensor. There is an active high interrupt which triggers when the infrared sensor detects an object. This interrupt will call the “detected” function in order to switch the system into its relevant state for having detected an object. When the infrared sensor stops detecting an object, it will trigger an active low interrupt in the system which will then call the “not_detected” function which sets the relevant variables back to their not detected state.

Solution Development

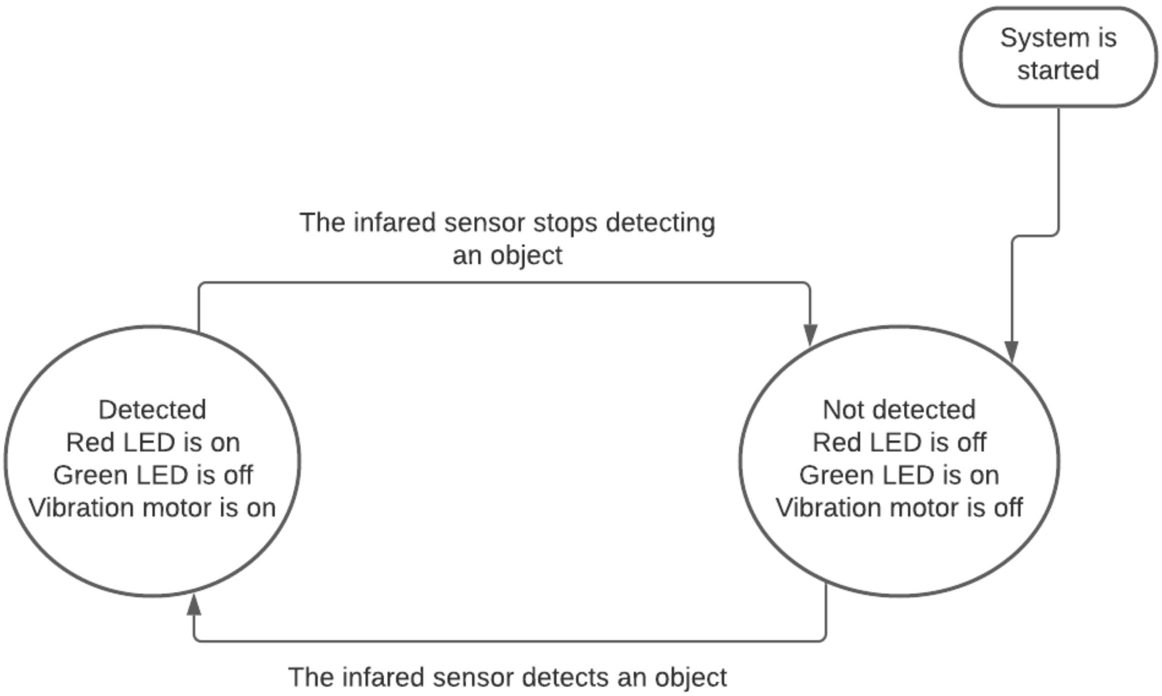
The project first started off the notion that an embedded system could provide feedback to the visually impaired. From there, the various sensors that were available were examined until it was determined that a vibration motor was most effective for feedback for the visually impaired, and ultrasonic and infrared sensors could both provide a form of object detection. LEDs were then added as feedback for the sensors in order to help with debugging, and because they could also be helpful for those who are not totally without sight. The vibration motor was found to only have two states, on, and off, and so the different inputs that the ultrasonic sensor could provide seemed unnecessary, so the infrared sensor was decided on.

Once coding was underway, there was an attempt to implement threads, one thread for inputs, the other for outputs, but MBED studio refused to acknowledge the declaration of a thread as valid. Hours were spent attempting to debug this, but to no avail, so the use of threads was abandoned. Interrupts were found to be the logical choice to connect the infrared sensor to the system outputs and were thus decided on. Finally, the vibration motor and infrared sensor were attached to an old parking permit for the simple reason that it was a lightweight and sturdy object, and from that, the design was complete! After the design was tested and seemed to work, a watchdog was added to make sure that the system did not get stuck in an error state.

System block diagram



Finite state machine

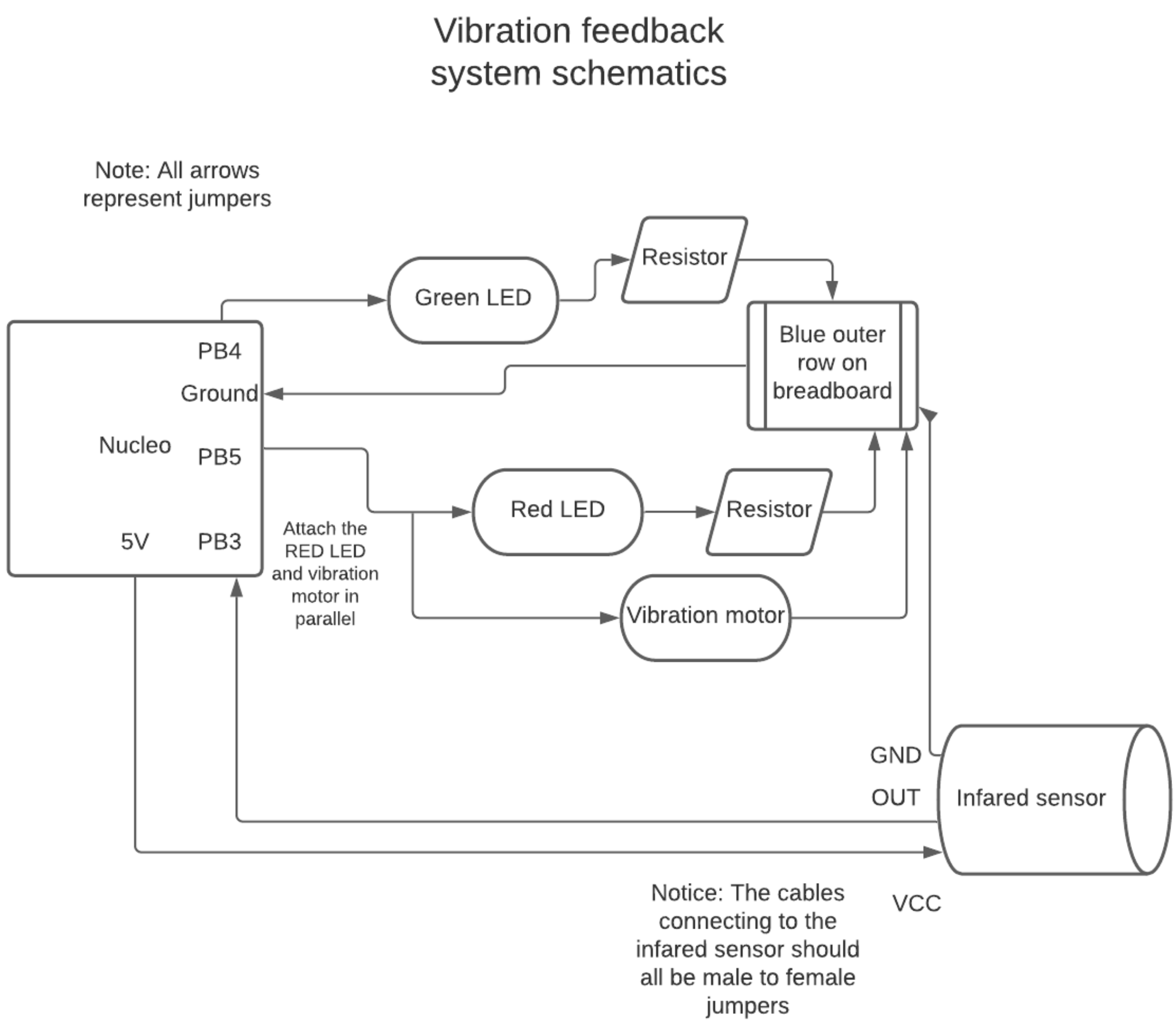


BOM

- A PC capable of running MBED studio
- Nucleo L4R5ZI
- A Micro USB to USB type A cable
- A computer running MBED studio
- Jumpers
- 5 Male to female jumpers
- Vibration motor
- Infared sensor
- A Breadboard
- Two LEDs, one red, the other green
- Two resistors
- (optional) A small piece of cardboard, thin sheet of plastic, or other flat lightweight surface that objects could be attached to.

User Instructions

Schematic



Instructions to build

1. Connect jumpers from the 5v port on the Nucleo to the red outer row on the breadboard,
2. Connect a jumper from a ground port on the Nucleo to the blue outer row on the breadboard.
3. In series, connect a jumper from PB4 to a spot on the breadboard, then connect a green LED to that, a resistor to the negative end of the LED, and finally connect the resistor to the blue outer rail of the breadboard which should be grounding.
4. Repeat step 2 but with a red LED instead of a green one, and connecting to PB5 instead of PB4.
5. Use three of the male to female jumpers to connect the infrared sensor's ground pin to the blue ground row on the breadboard, the VCC pin on the infrared sensor to the red power row on the breadboard, and the OUT pin on the sensor to PB3.
6. Connect a male to female to the row where PB5 connects to the red LED, and attach the female end to the positive end of the vibration motor.
7. Use another male to female jumper to finish the connection from the vibration motor's negative cable back to the grounding row on the breadboard.
8. (Optional) Attach the vibration motor and infrared sensor to the optionally acquired external surface specified in the bill of materials. This step is not necessary for the device to function, however it does create a convenient object to point at obstacles that's separate from the board that can both detect objects, and provide vibrational feedback.

Instructions to use

1. Ensure that the nucleo L4R5ZI is plugged into a computer and is running the provided code.
2. Test that connection are as they are supposed to be by placing an object in front of the infrared sensor. The red LED and vibration motor should turn on for the duration that the object is there.
3. If step 2 passes, the system should be ready. It can now be used to detect any objects in front of it.
4. Point sensor wherever an object may be and use the vibration motor for feedback. For the best effect when testing, close your eyes.

Test Plan

The following tests were developed in the order that they are listed in order to test the device as it was assembled.

1. LED check: This is a simple test that first turns on the green LED, then turns it off, then does the same thing to the red LED. This ensures that the LEDs are connected correctly if they turn on and off as expected.
2. Vibration motor check: This uses the same code as test 1, however it now also has the vibration motor connected to the same port as the red LED in order to test that it functions as intended and the motor is not dead. If the motor activates when the red LED does, the test passes.
3. Detection test: The infrared sensor has an object placed in front of it and removed. Each time an object is detected or ceases to be detected, a relevant print statement should be printed to the console.
4. Full system test: This test tests every component of the system by attempting to use the system as intended. If the LEDs and the vibration motor behave as described in the system over when the infrared sensor detects and then does not detect an object, the test passes.

Outcome of Implementation

Despite threads not being recognized, the implementation was a success. The vibration motor and LEDs trigger at the right time, and the system is responsive to such an extent that it is possible to use the system like a morse code device with the sensor acting as the button, and the LEDs and vibration motor acting as the output.

Future Considerations

The major shortfalls of the system are primarily in two areas. Firstly, the infrared sensor's range is very short. This would make the device less practical to use while walking as the user wouldn't have much of a response time when dealing with objects.
Secondly, the vibration motor has cables significantly smaller than the standard jumper size. This results in the motor often shaking its connecting cables free of whatever it is plugged into, whether that be the breadboard, the Nucleo ports themselves, or male to female jumpers.

There are several ways in which this implementation can be improved. Alligator clips attached to jumpers may prove a more effective method of keeping the vibration motor connected.

Alternatively, tape might help keep the cables of the motor connected to the jumpers. The object that has the sensor and motor on it would benefit greatly from being able to be moved farther from the breadboard and nucleo which might be achieved with something as simple as longer cables and a zip tie to keep said cables together. Lastly, the addition of a device that provides audio feedback in addition to the existing feedback system would be considered, though it would compromise the current relatively undisruptive nature of the system if implemented.