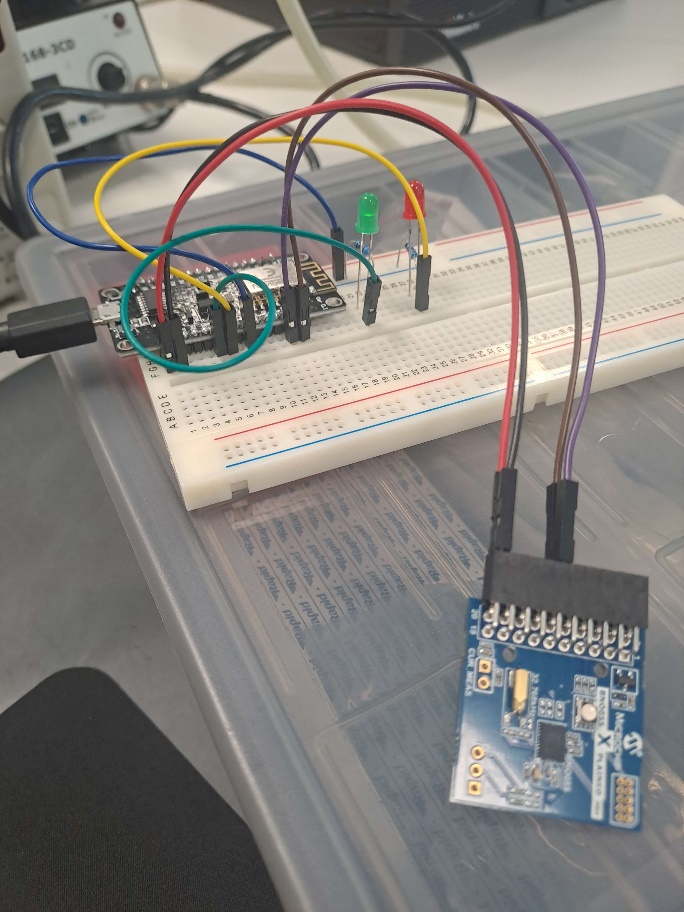
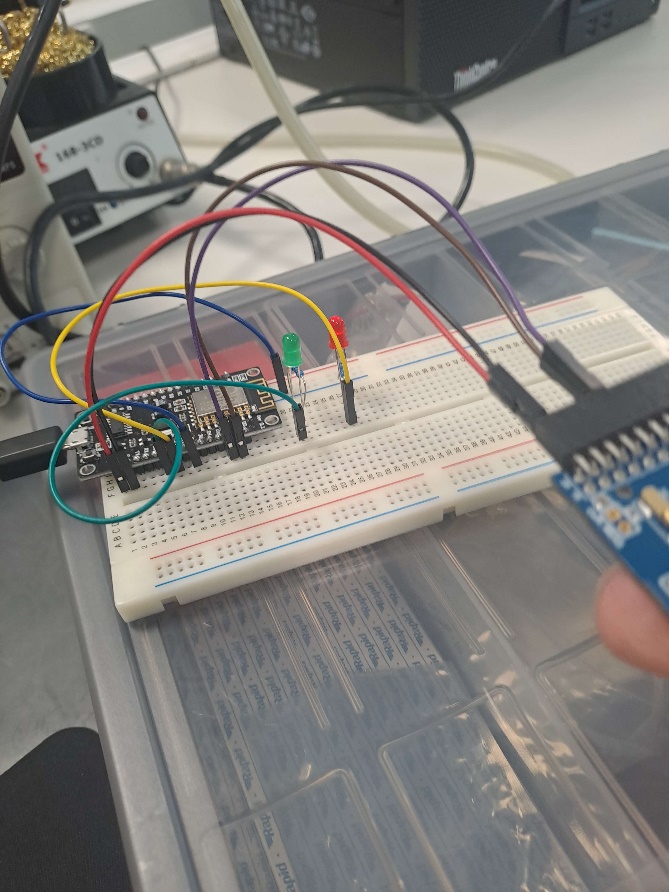
Overview

The Jewellery Box Motion Detection Module is a small device designed to be placed in a jewellery box to produce an alert if the box is moved. It works by using a BNO055 inertial measurement unit to report the orientation of the device. If the device’s orientation changes too rapidly, such as if the box were picked up, it generates an alert signal that is sent to the rest of the network, informing it of a potential theft.

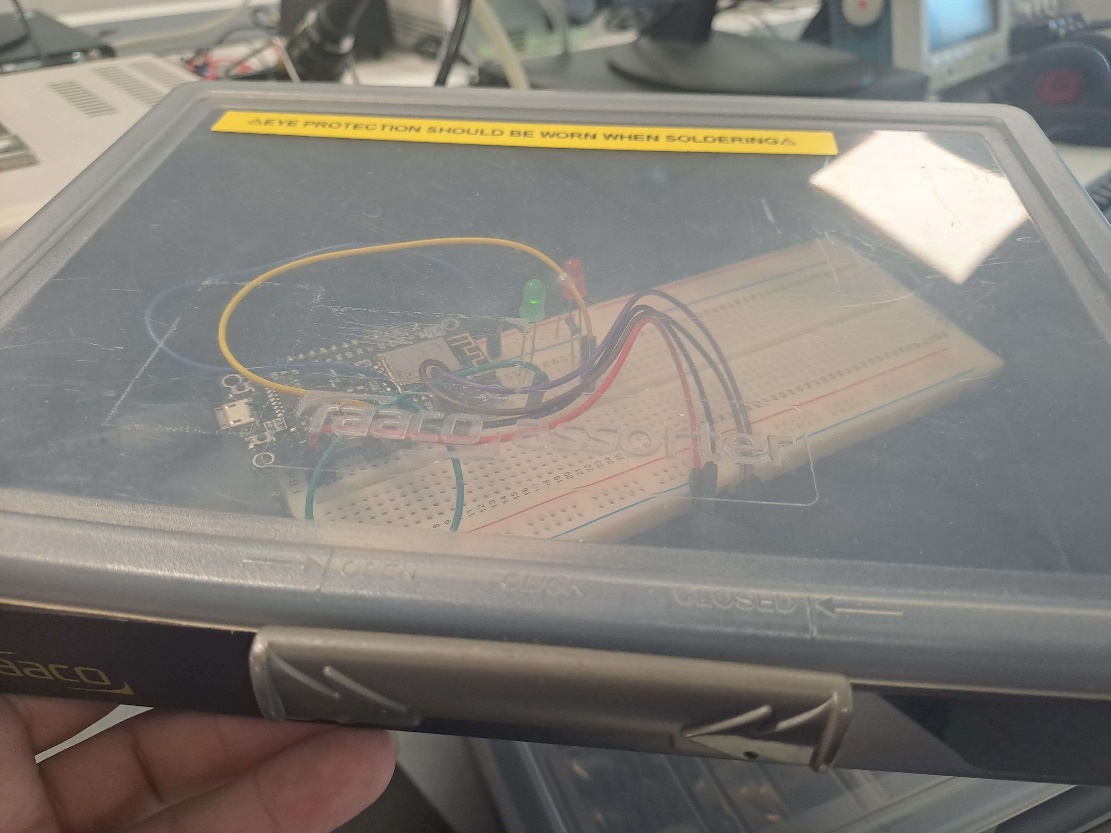
The BNO055 is run using Adafruit’s sensor libraries. An I2C communication line receives the measurements from the IMU, and the values are read in the form of a Quaternion, a vector type that combines the rotation values around the X, Y, and Z axes to produce a single value representing the rotation around an arbitrary axis known as the Euler axis. By having a single value that represents all rotation, it makes it easier to compare the data against a rolling average of previous readings, allowing the device to not produce false positives from small bumps to the surface the device is on, while also making it harder to move the device without setting off the alarm.

Functions

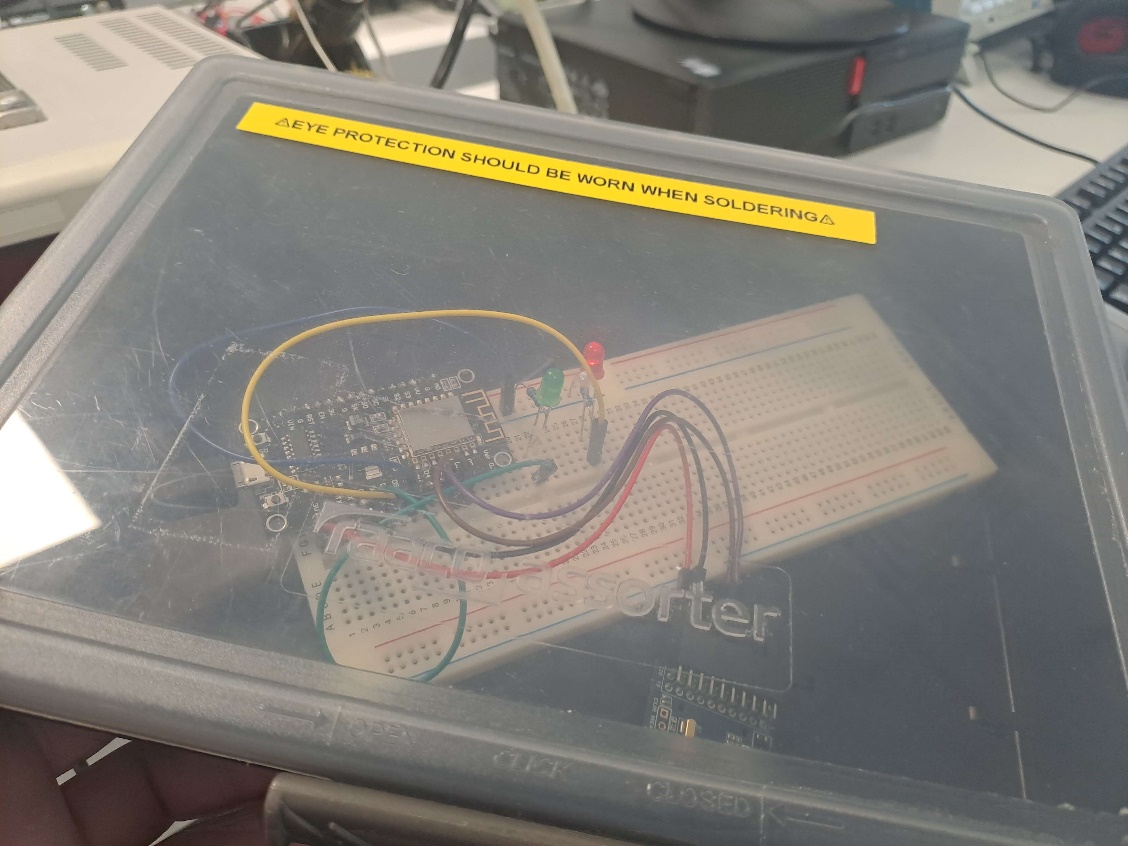
The module only has one function: to send an alert to the rest of the network if the module is moved. It is constantly comparing its current state to a rolling average of previous readings. If the new reading varies too much from that rolling average, an alert is generated, which is then sent out. During development, visual indicators were added to be able to test the system independently.

Device while stationary Device while IMU is being moved



Device inside a stationary box



A close up of a text

Description automatically generatedDevice inside a moving box

Sample of serial terminal output while device is stationary

A white paper with black text

Description automatically generatedSample of serial terminal output while device is producing alerts

A grey rectangular object on graph paper

Description automatically generated

Model of a box to place the device into

A grey box with a blue and black rectangular object

Description automatically generated

Model with the lid removed. Black box represents the microcontroller, blue box represents the IMU, and the silver cylinder represents a rechargeable battery

Challenges

The program required many iterations to reach its final state. First, the initial code relied on an outdated version of the sensor libraries, as it was recycled from an older project, and newer versions of the libraries had changed the formatting of some of the functions. The program also needed slight adjustments to account for the fact that it was originally written for use on Arduino boards, rather than an ESP8266.

Bugs

Placing the device in a box and attempting to lift it without producing an alert revealed that a steady hand could move the device vertically and horizontally without setting it off, though any attempt to rotate it would immediately generate an alert. This was because the device originally only considered its angular position when deciding whether it had been moved. By changing the way the device decides whether it has been moved to incorporate the accelerometers that the BNO055 has, a new threshold was able to be established that means it would be prohibitively slow for any would-be thief to avoid setting it off.

Occasionally, on initial booting of the system, the IMU would be unresponsive, only giving values of zero on all data points. To fix this, a system was added that would wait for a number of cycles of the main loop, then check if the data values were blank. If they were, then the program would automatically reset itself. While implementing this, a toggle was added to prevent the system from generating movement alerts while it was still calibrating its position.

Testing

Numerous tests were run, involving attempts to move the device (both inside and outside of a box) as well as bumping the table near it, to establish a suitable threshold for deciding whether changes in position were significant enough to trigger the alert. For ease of use during testing, a pair of LEDs and a serial communication port were connected to the device. By doing so, the device had visible ways of declaring an alert that did not rely on the presence of other nodes in the network.

Limitations

As the system is very simple at its core, the main limitations are its size and how long it can remain powered if it has to run off of a battery.

<Notes>

Integrating the systems to communicate with the rest of the network went smoother than expected. Other than the new functions needed for it, there was only two changes that needed to be made. First, the alert function needed to call the new function that sends messages to the network. Second, the function calls in the main loop had to be moved to a scheduled task, as the mesh system requires the main loop to only call the mesh’s update function. As much of the code that was previously in the main loop only served as a method to delay the device from producing alerts while the rolling average of its position calibrated, it could be blended into the setup function to remove an if statement that became unnecessary after the first second of the device running.

Running the function to read the IMU on a scheduled task introduced a delay in how often it was called, which had the unexpected side effect of increasing the device’s ability to tell the difference between negligible movement from bumping the table and significant movement that would justify generating an alert, as in the case of bumping the table, the device would return to its original position before the next call of the reading function.

The device could be improved by mounting it onto a permanent circuit board, rather than just a temporary breadboard. It also currently relies on being plugged into a usb port to receive power, which ideally should be replaced with a rechargeable battery.