Earthquake Detector Report

Members: Yongjun Kwon, Ho Yin Kam, Woojin Choi

Mentor: Brian Mellon

Aim:

Construct an earthquake detector that can detect an earthquake, output the corresponding magnitude of it using a combination of gyroscope and accelerometer data, storing the data digitally, and alert people in the surrounding autonomously.

Abstract:

Researches have been conducted that ants were able to detect an earthquake with the change in gas emission (Hydrogen and Rodon) or by electromagnetic waves. We delved into this topic for further information and how the detection mechanism works. Although there are few significant papers regarding this subject, there are not enough data to reach a conclusion of such mechanism and the research is not able to show a cause and effect relation between gas anomaly and earthquake yet (only correlation has been illustrated). While the research on ants may not be statistically significant for our earthquake detection use, we were inspired by another research conducted by UC Berkeley about using mobile devices to create a distributed network for earthquake detection. As a result, we have resorted to using accelerometer and gyroscope combination to determine the presence of earthquake as we find this way more accurate, cost effective, and intuitive.

Apparatus:

Hardware and Electronics:

- Arduino Uno
- Arduino Nano
- MPU6050 DMP6 (3-Axis Accelerometer and 3-axis gyroscope)
- GROVE buzzer
- 3 LED light (Red, Yellow, and Green)
- Always-off Vibration Switch (SW-420)
- Real Time Clock Module (DST3231)
- SD Card Module
- Bluetooth Module (HC-05)
- Wooden Case (13 cm by 14 cm by 10 cm)

Software:

- Arduino Nano software that saves data to sd card and transmits data to computer through Bluetooth module
- Arduino Uno software that retrieves data from the MPU6050 and associates the data with the actual data retrieval time using a real-time clock module
- Website in javascript that displays the data in graph form.

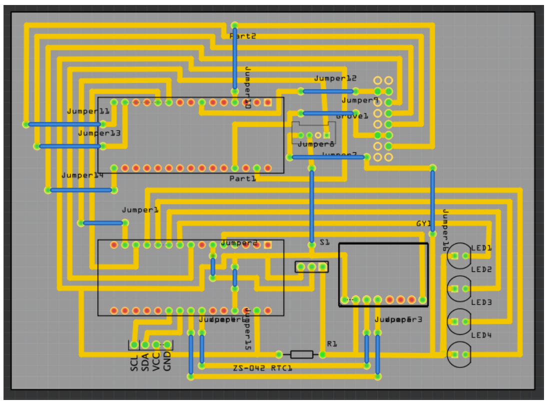
Documentation/Procedure:

A working prototype is shown in the following link: https://drive.google.com/a/ais.edu.hk/file/d/0B9tS-JHjEeC7NWtzNWlVLTgwYlU/view?usp=drivesdk

Electrical Design:

We had to use different hardware materials to finish this earthquake detector and used the Nano Arduino as the mainframe to act and gain data of various components of the machine.The program made for the nano to get the gyroscope/accelerometer was to serve as a motion detector and data to get the roll, yaw, and pitch. The Real Time Clock was used to get the difference in time as milliseconds. The vibrator switch detected if there was a vibration and triggers the buzzer and the LEDs to turn on if the vibration is in accordance with the accelerometer and gyroscope sensor. (i.e. if there is a spike in gyroscope data, it may be some noise or interference. But if we combine with a vibration switch, we can clearly see if the data is a false alarm or not.)

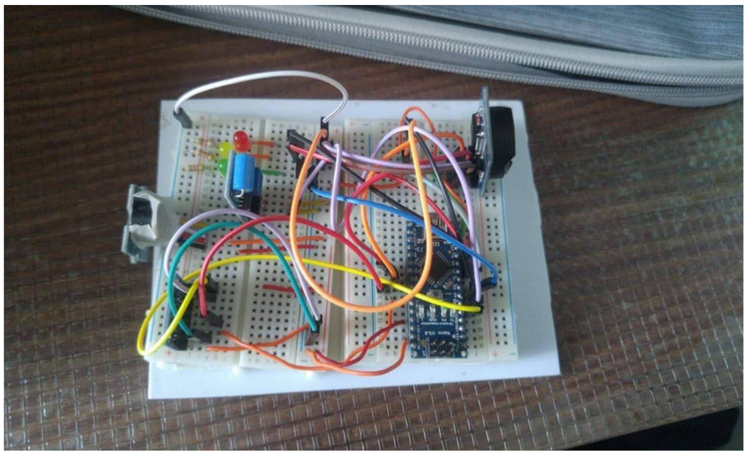
Circuit Board Design:



We have designed a circuit that integrates our whole system into one unified board. This way allow us to organise our connections better and also prevents unexpected errors, such as malfunctioning and loose wires. The top right is the SD card component, connecting to the top Arduino. The bottom right consists of 4 LEDs, which are indicators for power, vibration switch, gyroscope/accelerometer, and earthquake. Next to the LEDs is the MPU 6050, which is connected to the bottom Arduino. The bottom Arduino is attached to the Real Time Clock right below it. After every iteration of code, the data will be compiled and send to the top Arduino using Software Serial using port 10 and 11 and a common GND. The buzzer is connected to the bottom Arduino, so when our program concluded that there might be an earthquake, it will be activated.

Mechanical Design:

We have decided to make a wooden box to put the earthquake detector in because we wouldn't want the machinery to break. The box was made in Maker Bay, a company that sponsored our team. With their contribution, we are able to make a box with their tools for the earthquake detector. All components in the earthquake detector will be placed in the box except the LEDs. With that in mind, we construct a 13x14x10 centimetre wooden box. We made the box by cutting wooden pieces with a blade saw, installing angle connectors, adjusting the box to the right size and sandpapering the box.

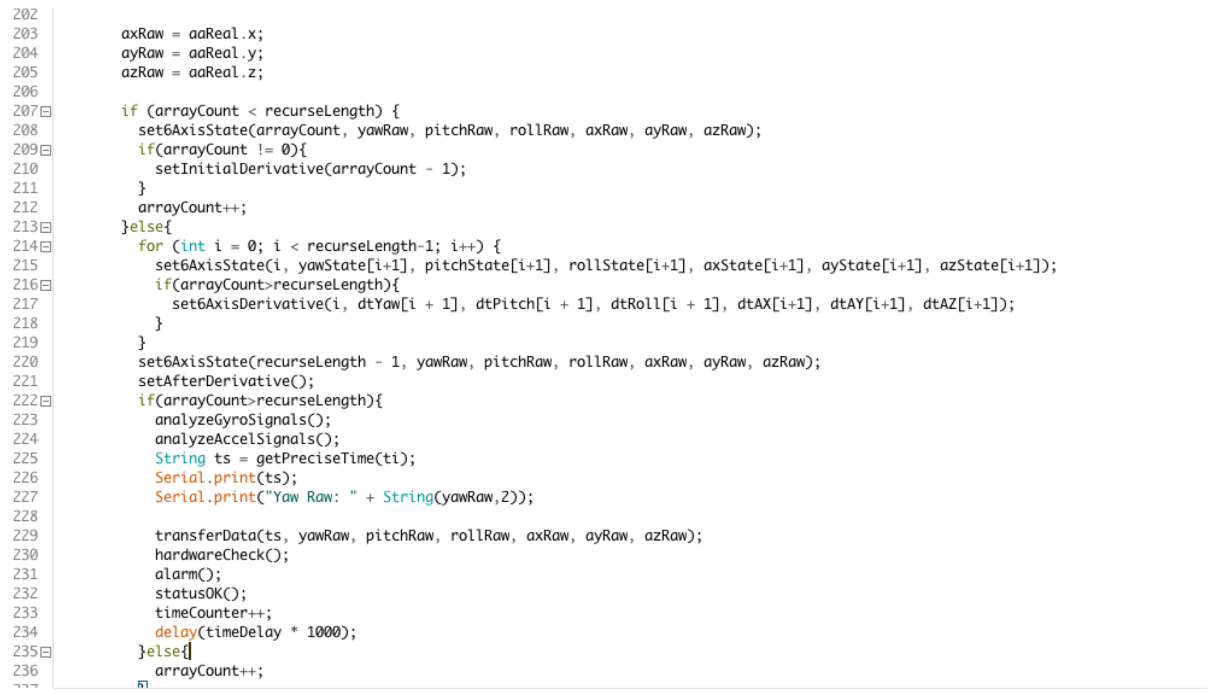


Software Design

The software is separated into 2 section: the data retrieval and compilation, and the data storing and sending. Mentioned above, we are using two Arduino in our detection system. The two Arduino are connected for data transmission in this following format (time, yaw, pitch, roll, accelerometer x,y,z, average derivative for yaw, pitch roll, x,y,z, magnitude calculated from gyroscope data, magnitude calculated from accelerometer data). The reason we have to separate the system into two Arduino is that Arduino nano does not have enough processing power and ram to run all the calculations regarding the earthquake magnitude, and not enough space to store all the data in our 12 arrays, each storing 12 data entry. Also, even if the board is switched to Arduino Uno, the calculation, data retrieval, and time retrieval take around 0.01-0.03 milliseconds to process, which creates a lot of trouble when we try to log our data as the time will be accumulatively more inaccurate. To transfer data from the Arduino UNO to Arduino nano, we used an internal protocol called the Software Serial, allowing us to transmit around 100 bytes of data every 0.08 seconds. We choose this over I2C for data transmission because I2C has an inherent limitation of 32-bytes of data transferal.

Data retrieval and compilation

We retrieve the data from MPU 6050 using the standardized I2C protocol. When the sensor is first initialized, it is really unstable, so we have devised a code to check the stability of the sensor. If the sensor is stable, the program then continue to retrieve data from our sensors and real time clock; If not, the program will keep looping until the sensor reaches data stability. During data retrieval, we store our data gyroscope yaw, pitch, roll, accelerometer x,y,z, into 6 arrays of length 12. We then calculate the derivative (changes) in between each previous data points, and generate another 6 arrays of length 12 that stores all these calculated derivatives. All these data will be passed through a function that calculates the average value of the array and this value will be sent to another function that determines the magnitude of the earthquake based on our empirically defined constants (shown in "How to measure data and calculate magnitude" section). If this value passes a certain threshold that we define, and this spike of change corresponds to data collected from the vibration switch, we will conclude that there is an earthquake and activate the LEDs and buzzer. A picture of the core algorithm of the program is shown below.



```

    axRaw = aaReal.x;
    ayRaw = aaReal.y;
    azRaw = aaReal.z;

    if (arrayCount < recurseLength) {
      set6AxisState(arrayCount, yawRaw, pitchRaw, rollRaw, axRaw, ayRaw, azRaw);
      if(arrayCount != 0){
        setInitialDerivative(arrayCount - 1);
      }
      arrayCount++;
    }else{
      for (int i = 0; i < recurseLength-1; i++) {
        set6AxisState(i, yawState[i+1], pitchState[i+1], rollState[i+1], axState[i+1], ayState[i+1], azState[i+1]);
        if(arrayCount>recurseLength){
          set6AxisDerivative(i, dtYaw[i + 1], dtPitch[i + 1], dtRoll[i + 1], dtAX[i+1], dtAY[i+1], dtAZ[i+1]);
        }
      }
      set6AxisState(recurseLength - 1, yawRaw, pitchRaw, rollRaw, axRaw, ayRaw, azRaw);
      setAfterDerivative();
      if(arrayCount>recurseLength){
        analyzeGyroSignals();
        analyzeAccelSignals();
        String ts = getPreciseTime(ti);
        Serial.print(ts);
        Serial.print("Yaw Raw: " + String(yawRaw,2));

        transferData(ts, yawRaw, pitchRaw, rollRaw, axRaw, ayRaw, azRaw);
        hardwareCheck();
        alarm();
        statusOK();
        timeCounter++;
        delay(timeDelay * 1000);
      }else{
        arrayCount++;
```

Data storing and sending

This code is on the top Arduino, running recursively to retrieve all the raw data and calculated derivative arrays from the bottom Arduino. We will first run a preliminary check on the data to make sure the data received are accurate and not some random signal noise. To ensure the integrity of data, we send it in a particular format so that when the program checks the pattern and if it doesn't match, that whole iteration of data entry will be disregarded in our storing process. These data will be compressed and reduced before storing because according to our calculation, we would need an 8gb SD card every month as it generates approximately 300 MB per day. We reduced the data only to have the raw data values and the time of data entry as the derivatives and magnitude can be calculated on-the-fly as the computer crunches through these massive amounts of data. Through this, we can reduce approximately 60% of the data sent, thus profoundly improving the usability of our system. After storing data, these entries will be broadcasted to the surrounding through Bluetooth using the HC-05 module. A computer can read these data in real time through its internal Bluetooth serial and make use of these real-time data (such as creating a graph of the data or instantaneously send warning to family members).

How to measure data and calculate magnitude:

A demo of the data calculation is shown here: https://drive.google.com/file/d/0B9tS-JHjEeC7X1VtSnVrNmxDQzg/view?usp=sharing

We measure our data using the MPU 6050 sensors. After data is received and processed (calculating derivative array and average array value), we calculate the earthquake magnitude using the equation:

$$average\ derivative = ke^{c(magnitude)}$$

We can rearrange the equation and calculate the average magnitude from our data by changing the formula to:

$$\frac{average\ magnitude}{k} = e^{c(magnitude)}$$

$$ln(\frac{average\ magnitude}{k}) = c(magnitude)$$

$$\frac{ln(\frac{average\ magnitude}{k})}{c} = magnitude$$

We devised this equation based on the idea of Richter scale, a logarithmic scale for quantizing the strength of earthquake. We used a combination of some test data and online researches to determine exponential regression line, and use it to find the constants c and k for our equation.

Conclusion

The prototype shows a new and efficient way to detect the earthquake. We used the gyroscope/accelerometer because it is a much efficient way to make data than that of seismographs. We would be able to use a quake alarm where they alarm when an earthquake is coming. However, it can only approximate the magnitude of earthquake based on our limited testing. Based on this approximate magnitude, we can thus estimate how much damage it had caused. This earthquake detector is also economical to build. Researchers has used laser beams and different instruments for the construction of such detector, which are expensive, yet our whole system has only cost us a mere 156 HKD.

Cost:

Item	Cost
Arduino Nano/ uno	100 HKD
Buzzer	5 HKD
Accelerometer	8 HKD
4 LED light	1 HKD
SD card	2 HKD
Real Time Clock	6 HKD
Bluetooth Module	20 HKD
Vibration Detector	8 HKD
Circuit board	6 HKD
Total	156 HKD

Acknowledgement:
We would like to give special thanks to the following people/ company who have supported us during the course of this competition.

Eunsup Kang

Brian Mellon

Justin Bae

MakerBay

Reference:

December 07, 2010 | 20 Comments. "Earthquake Data Logger." Earthquake Data Logger - SparkFun Electronics. N.p., n.d. Web. 24 Apr. 2017 <.https://www.sparkfun.com/tutorials/235>.

"Earthquake Indicator Using Arduino - Page 2 of 2." Electronics For You. N.p., 02 Feb. 2017. Web. 24 Apr. 2017. <http://electronicsforu.com/electronics-projects/hardware-diy/arduino-earthquake-indicator/2>.

Veedo. "Arduino Seismic Activity Monitor - Ethernet Shield." Instructables.com. Instructables, 12 May 2016. Web. 24 Apr. 2017. <http://www.instructables.com/id/Arduino-Seismic-Activity-Monitor-Ethernet-Shield/?ALLSTEPS>

"Measuring the Size of an Earthquake." U.S. Geological Survey. N.p., n.d. Web. 24 Apr. 2017. <https://earthquake.usgs.gov/learn/topics/measure.php>.

"Hong Kong Observatory." Modified Mercalli Scale. N.p., n.d. Web. 24 Apr. 2017.<http://www.hko.gov.hk/gts/equake/mms_e.htm>.

Berberich, Gabriele, Martin Berberich, Arne Grumpe, Christian Wöhler, and Ulrich Schreiber. "Early Results of Three-Year Monitoring of Red Wood Ants' Behavioral Changes and Their Possible Correlation with Earthquake Events." MDPI. Multidisciplinary Digital Publishing Institute, 04 Feb. 2013. Web. 24 Apr. 2017. <http://www.mdpi.com/2076-2615/3/1/63/htm>.

"Can Ants Predict When an Earthquake Will Strike?" Local Weather from AccuWeather.com - Superior Accuracy™. N.p., n.d. Web. 24 Apr. 2017. <http://www.accuweather.com/en/weather-news/can-ants-predict-when-an-earth/10211815>.

Oskin, Becky. "Ants Lead the Way on Earthquake Prediction." N.p., 11 Apr. 2013. Web. 24 Apr. 2017. <http://www.livescience.com/28672-ants-sense-earthquakes.html>.

Zöller, Gert, Sebastian Hainzl, and Jürgen Kurths. "Observation of growing correlation length as an indicator for critical point behavior prior to large earthquakes." Journal of Geophysical Research: Solid Earth. N.p., 10 Feb. 2001. Web. 24 Apr. 2017. <http://onlinelibrary.wiley.com/doi/10.1029/2000JB900379/full>.