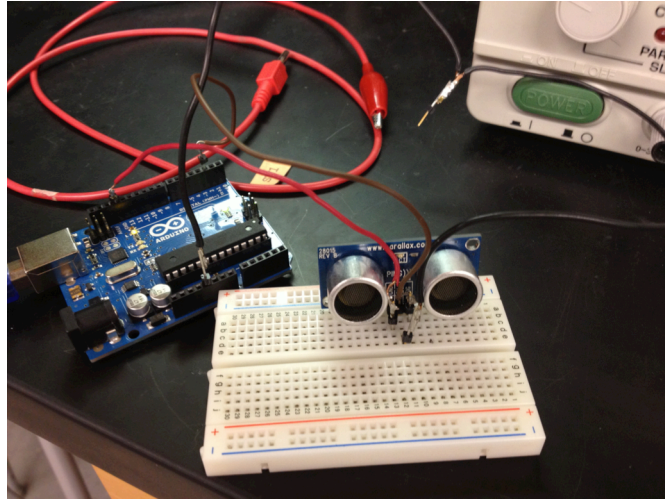*Student Name: Jeremy Lange*

Project Title: Mail Detection System

Description:

One problem that arises regularly is the worry of getting to mail on-time, whether it is to get bills in a timely manner or getting a package that has been anxiously waited on. Another problem, one that I personally wanted to combat having experienced it first-hand, was the fact of others rummaging through my mail and taking what they please. The biggest obstacle to overcome was finding some way of notification when the mail was delivered or when the mailbox was opened without approval.

To solve this, a simple motion detection system was needed although this only solved part of the problem. I chose to have the arduino send a text message anytime a door opening was detected. This part took some work when using only an arduino and no GSM board as an attachment. To simplify this project, it was assumed that the arduino would be connected via USB to a computer that is connected to the Internet at all times.

The benefit to having this system implemented is knowing exactly when mail is delivered everyday as well as being able to tell if someone is snooping through post without authorization.

Inputs:

In this system, the input is taken from the parallax ping))) ultrasonic sensor. This sensor sends out a high frequency signal that is reflected by some obstruction in front of the sensor. The soundwave then bounces off of the object and is returned to a microphone that listens for the return signal. The ping)))

measures distances by outputting a time in ms. From this reading a distance can be found by using the formula:

$$distance = \frac{time}{73.4 * 2}$$

The total time is divided by 2 since the signal has to first reach the object and then return the same distance. It is also divided by 73.4 because according to parallax's website, sound travels at a rate of 73.4 ms per inch.

An assumption was made that the typical mailbox is roughly eighteen inches in length. Performing the experiment in the lab while printing out measured time from the ping))) sensor, it was found that a distance of eighteen inches corresponded to a printed time of roughly 20,000 labeled as "duration". Within the void loop, and if else statement was constructed so that if a time of 20,000 or less was measured it was assumed that the mailbox door was closed and there was no motion and a 0 was printed. If a measurement was over 20,000 it was assumed that the mailbox door had been opened and a 1 was printed signaling motion.
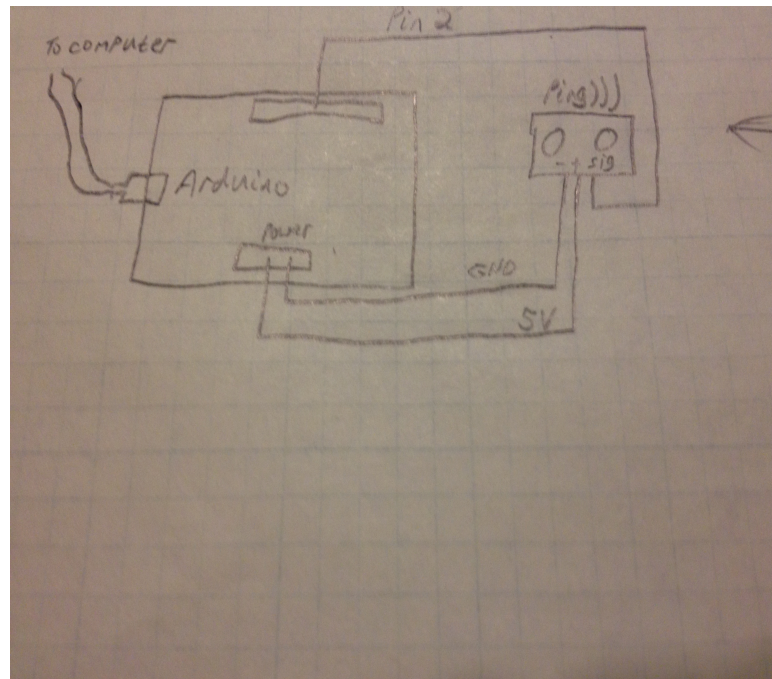
Outputs:

The output of this system was essentially a printed 1 or 0 corresponding to "motion" or "no motion" respectively. A python script was compiled that constantly polled the arduino at a specified interval, one second in this case, and looked for the 1 or 0 to correspond to an output. If a 1 was received from the arduino, the python script then sent a text message to a specified cell-phone number in the variable section of the script using an API from the service provided at www.telapi.com. The LED located on pin 13 of the arduino also lit up when a 1 was received to signal that the input was registered. If a 0 was received, the LED stayed off and the python script continued polling the PLC. The python script will not send a text message more often than the specified interval allows, 2 minutes in this case, in order to avoid a constant "spam" of messages to the receivers cell phone.

Functions:

The function of this system is defined below:

1. The ping))) sensor sends out a frequency to measure the distance of some object placed directly in front.
2. When the return time is over a specified number of 20,000 it is assumed there is motion taking place in the system and a 1 is printed to signal this.
3. The python script registers the 1 as motion detected and contacts www.telapi.com using specified parameters in order to send a text message to the intended receiver.

Design:

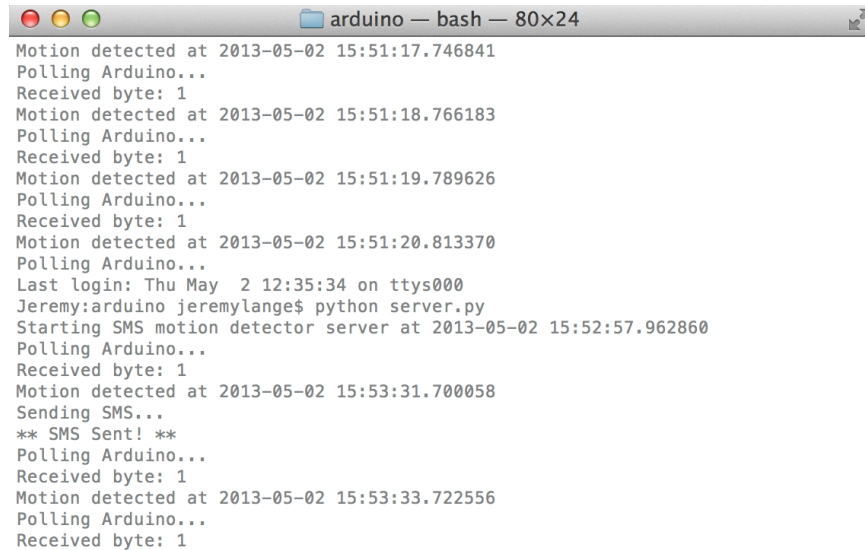
Measurements and Calculations:

While parallax, the manufacturer of the ping))) sensor, gave the formula to find distance from the time read when printed from "duration" I was unable to get it to match the calculated result for 18 inches. The calculated value came out to be 2644 using the equation listed under the above section "inputs" which is off by almost a factor of 10. This could have been due to that fact of external noise in the lab being picked up by the ping))) sensor as well as the objects being placed in front being rounded and not straight.

Results:

Testing this system, multiple oddly shaped objects were placed in front of it at various distances. The results were favorable and anything up to seventeen inches always registered a 0 printed from the arduino. Measurement error arises from other frequencies entering the microphone, which is a downfall of using an ultrasonic sensor versus a photoresistor to measure motion.

Discussion:

The performance of this system worked exactly as planned aside from the discrepancy between the hand calculated conversion of time to distance and what the arduino actually measured. When an object was placed eighteen inches or less in front of the arduino, no text message was sent. As soon as the object left the specified range or nothing was placed in front at all, a text message was relayed to signify that "motion" had been detected.

As seen from the two figures above, when running the python script from terminal using Mac OSX, as the arduino was polled for data if a 1 was detected then a text message was dispatched. The telephone number the automated text messages were sent from was registered from telAPI as (224) 220-1228.

Improvements and Extensions:

Improvements that could be made to this design are more refined ways of measuring distance, as well as an input device that measures light versus sound. An input source such as a photoresistor would improve accuracy of this PLC as it would not pick up any ambient noise created by pedestrians, cars, or any other animal passing by. Another improvement that would help users that have mailboxes away from their home would be the incorporation of solar or battery power as well as a wifi board attached to the arduino, which would eliminate the need to be plugged in via USB to a host computer.

References:

https://github.com/mattwilliamson/arduino-sms-alarm

http://arduino.cc/en/Tutorial/Ping