Characterization of Programmable Arduino Sensors

Brett Mortenson
Utah State University

Follow this and additional works at: https://digitalcommons.usu.edu/phys_capstoneproject

Part of the Physics Commons

Recommended Citation
https://digitalcommons.usu.edu/phys_capstoneproject/63

This Report is brought to you for free and open access by the Physics Student Research at DigitalCommons@USU. It has been accepted for inclusion in Physics Capstone Project by an authorized administrator of DigitalCommons@USU. For more information, please contact digitalcommons@usu.edu.
Characterization of Programmable Arduino Sensors

Brett Mortenson
Jonathan Price, Project Mentor

Department of Physics, Utah State University, Logan, Utah

Arduinos are increasing in usage and prevalence due to the ease in which they can be programmed. External sensors attached to the Arduinos are being used for various measurements, and many of them give raw data output such as voltage spikes or decibel readings. These readings were characterized so that one can interpret the readings relatively easily, and understand more completely, the environment which the sensors are in. The sensors utilized are as follows: photoresistor, vibrational, sound, temperature, pressure, and humidity.

Experiment Date: September 2017-April 2018
Introduction

In many instances, the Arduino sensors measure the environment in raw data which has no meaning to the casual observer. For instance, the vibrational sensor reads in voltage as well the photoresistor also records data in voltage. A voltage spike of 3V has little meaning unless you know what sort of vibration or light exposure can cause a 3V spike. The same principle goes for the sound detector, which records data in decibels. Different conditions were presented to the various sensors, and the readings were mapped their respective circumstances. This serves to give a standardized interpretation of the raw data. The temperature, pressure, and humidity sensors gave readings in standard units. These readings were compared to other known standards, measuring the same conditions, to determine the Arduino sensors accuracy.

Methods

The photoresistor was calibrated first. The photoresistor is light sensitive. When light shines on the resistor, the photons excite bound electrons which then jump up energy levels. This in turn allows, the now free electrons, to conduct electricity which lowers the resistance between the two terminals. Thus, the resistance of the photocell decreases, as well as the resistor which was connected to it. Due to Ohm’s Law (V=IR), this means that the current flowing through both resistors increases, which in turn causes the voltage across the 2.2KΩ resistor (the one connected to the photoresistor) to increase. The photocell was exposed to different levels of light; A dark room, a dimly lit room, and a bright room. The Arduino uses analog to digital conversion to display the resistance and voltage. The way this works is that the analog voltage charges up an internal capacitor, and then measures the time it takes to discharge across an internal resistor. The microcontroller monitors the number of clock cycles that pass before the capacitor is discharged.
The system is 5V, which has a resolution of 1023. This is written into the code as
\[
\text{lightV} = \frac{\text{lightADC} \times \text{VCC}}{1023.0} \quad (1)
\]
to read output as voltage. The voltage reading at these levels were then mapped respective to the varying levels of light.

Next was the vibrational sensor. This sensor works by utilizing the Piezoelectric effect. This phenomenon occurs in certain substances in which an electric charge accumulates in response to mechanical stress. In the vibrational sensor, the mechanical stress put on the sensor is due to a bending of the sensor due to vibrations. The same analog to digital conversion is used as in the photoresistor. The equation/code is:
\[
\text{piezoV} = \frac{\text{piezoADC}}{1023.0} \times 5.0 \quad (2)
\]

Weights of 500, 200, 50, and 20 grams were dropped from 2.54 cm, at a distance of 75 cm from the sensor, onto the desk the sensor was on. The corresponding readings were then mapped to their respective potential energies.

\[
\text{PE} = \text{mgh} \quad \text{where PE is the potential energy, m is mass in kg, g is gravity, and h is the height dropped.} \quad (3)
\]

The potential energies of the weights serve as a reference frame for the vibrational voltage readings. The angle, position of the weight on impact, and size of the weight were not included in this calculation.

Following the vibrational sensor, the sound detector was programmed. The
attempt was to calibrate the sound detector to a standard decibel meter. A program was found online in which a speaker emitted a known decibel level. The emitted sound level was tested against the decibel meter and the Arduino sound detector to try and find the relative accuracy of the sensor.

The T5403 sensor was used to measure temperature. It was placed in an insulated environment with ice to get an idea of the accuracy of the sensor beyond the given error analysis.

The pressure sensor was the second to last sensor calibrated. This sensor was compared to Utah State University's weather monitoring station to determine if the initial conditions coded in were accurate. The sensor has built into it the pressure equation. [2]

\[
\text{altitude} = 44330 \times \left( 1 - \left( \frac{p}{p_0} \right)^{\frac{1}{5.265}} \right)
\]  

(4)

In this equation, the altitude is the altitude at which the measurements are being taken, \( p_0 \) is the pressure at sea level (1013.25 hPa), and \( p \) is the pressure at the current altitude. In the code, the average altitude of Logan, Utah was used (1458 meters). Because altitude=1458 meters, and \( p_0 = 1013.25 \text{hPa} \), the sensor can then solve for \( p \), which is our output of pressure at the current altitude. The sensor was placed on campus at three different occasions. It was given 30 mins to adjust to the new environment and the pressure was then measured and compared to Utah State’s readings.

The humidity sensor was measured using the same technique as the barometric sensor.
Results

<table>
<thead>
<tr>
<th>Ambient light like...</th>
<th>Ambient light (lux)</th>
<th>Photocell resistance (Ω)</th>
<th>Voltage across R</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dim hallway</td>
<td>0.1 lux</td>
<td>600 KΩ</td>
<td>0.1 V</td>
</tr>
<tr>
<td>Moonlit night</td>
<td>1 lux</td>
<td>70 KΩ</td>
<td>0.6 V</td>
</tr>
<tr>
<td>Dark room</td>
<td>10 lux</td>
<td>10 KΩ</td>
<td>2.5 V</td>
</tr>
<tr>
<td>Dark overcast day</td>
<td>100 lux</td>
<td>1.5 KΩ</td>
<td>4.3 V</td>
</tr>
<tr>
<td>Overcast day</td>
<td>1000 lux</td>
<td>300 Ω</td>
<td>5 V</td>
</tr>
</tbody>
</table>

Fig. 2 (above): This chart correlates voltage displayed across the 2.2 KΩ resistor to the level of light the photoresistor is exposed to. [3]

The voltage readings across the resistor were matched to Fig. 1, giving us a solid approximation of the level of light the photoresistor is experiencing.

For the vibrational sensor, the voltage readings from all four different weights were recorded along with a corresponding potential energy. Any spike higher than 3.5-4 is considered a major vibration such as an earthquake or being placed on a lawn mower. It was deemed unnecessary to measure vibrations of such magnitude. Any voltage readings can be compared to Fig. 2. If you have a 3 V reading, then you have an idea the level of vibration experienced (e.g. potential energy of .02 J from a weight, dropped onto a standard desk 75cm from the sensor).

<table>
<thead>
<tr>
<th>Voltage</th>
<th>Potential Energy (J)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.36</td>
<td>0.049</td>
</tr>
<tr>
<td>3.12</td>
<td>0.019</td>
</tr>
<tr>
<td>2.25</td>
<td>0.005</td>
</tr>
<tr>
<td>1.11</td>
<td>0.002</td>
</tr>
</tbody>
</table>

Fig. 2 (left): Shows the vibration sensors voltage reading with the accompanying potential energy of the weight dropped 75cm from the sensor.
The sound sensor was not sensitive enough to be calibrated accurately. When compared to a decibel meter, it would consistently fail match it, or mismatch it in a way that was unidentifiable. It appeared that the sensor was not able to detect varying levels of sound, but that it was able to measure a certain sound threshold. The sensor was programmed such that it will register higher decibel levels with an output display of “high”. (e.g. shouting in the same room as the sensor, or loud clapping).

The temperature sensor read 32 ± 1 degree when kept next to melting ice. When compared to USU’s monitoring station, there was minimal variation. According to EPCOS, the temperature error is ±1 °C between 0 and 70 °C. The pressure error is ± .14 hPa between 500-1100 hPa and 25-40 °C. [4] The humidity sensor has a listed error of 2%. [1] USU has listed ±.03 kPa as the error for their barometric sensor, ±1°C for the temperature reading, and ±1.7% for humidity. [5] The Arduino was consistently within the margins of error upon comparison.

Conclusions

These experiments serve as a basis for understanding readings from the various sensors. It would be useful in the future to do multiple runs through each of the experimental setups. The data for each run would be recorded and the error for each sensor could be reported with the readings. This would serve to increase the accuracy and interpretation of Arduino sensors. Although this was not done, the characterized sensors are still useful in determining relative conditions and abnormal circumstances presented in a system. Also, it could be useful to compare identical sensors in the same experimental setups to see if they read similar results to the original sensors.
References