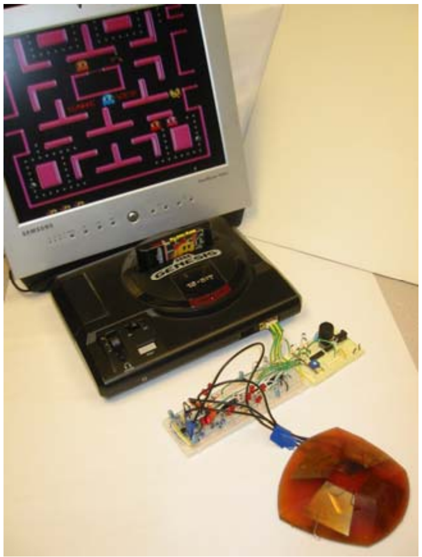
Capacitive Sensing Control

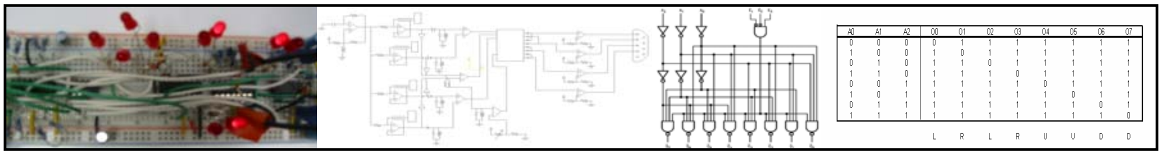
Will Lark
May 24th, 2007

Control anything with a simple wave of the hand… That is the focus of my capacitive sensing project. Initial plans focused on unique drive-by-wire interfaces; nevertheless, the overall goal of the project was to create a capacitive controlled virtual joystick. For the purpose of demonstration, I decided to retrofit the controller to an old Sega Genesis Pac-Man game. Although future designs would incorporate more sophisticated microcontroller managed components that would better decipher directional gradients, I felt that for the scope of this project and my limited electronics skills that creating a simpler sensor that deciphered between up-down-left-right in a relatively small package would suffice.

Beginning the project I had a couple of initial technical goals. Understanding that capacitive sensing can many times be flaky, the initial goal was to create a fairly robust sensor that would not be drastically affected by variations in environments and would not require the user adjust control behaviors. Also, although I believe I could have made this whole project physically simpler containing reasonably fewer parts, I planned to create the whole sensor from analog components to strengthen my understanding of analog circuitry (plus it is more fun to hack analog parts than program). Last, I planned to have the control sensitive enough that waving you hand a half a foot above it would activate it accordingly.

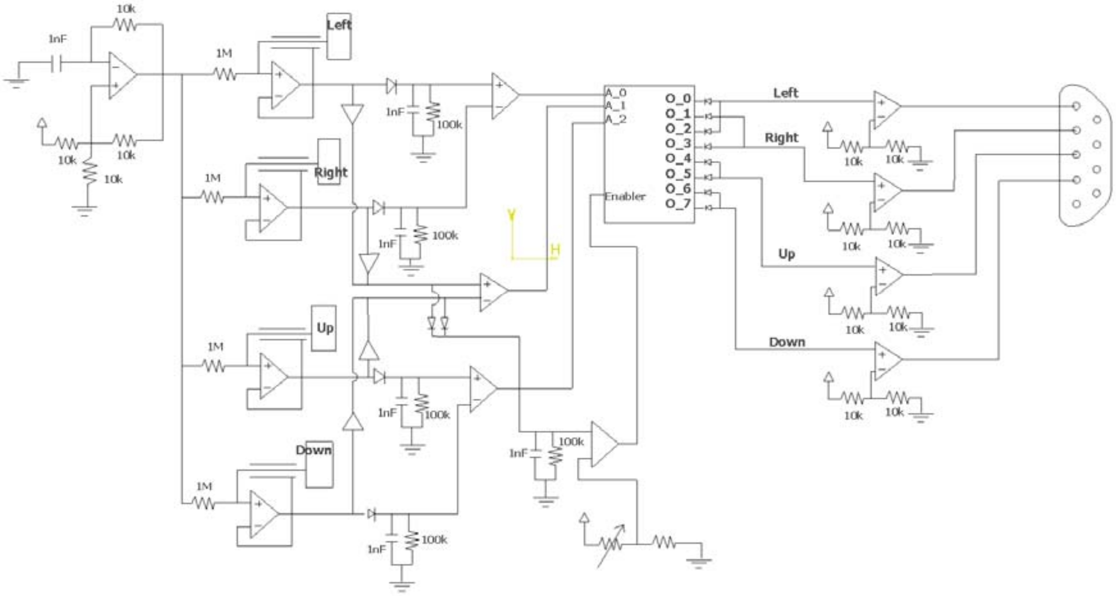


Design/Circuitry



A0	A1	A2	O0	O1	O2	O3	O4	O5	O6	O7
0	0	0	0	1	1	1	1	1	1	1
1	0	0	1	0	1	1	1	1	1	1
0	1	0	1	1	0	1	1	1	1	1
1	1	0	1	1	1	0	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1
1	0	1	1	1	1	1	1	0	1	1
0	1	1	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	0
			L	R	L	R	U	U	D	D

The circuit design was mainly composed of four appropriately positioned capacitive plates and analog circuitry that deciphered which value was larger by process of elimination. Also, even though each of the four (up down left right) capacitive first stage components are built identically, there are some tolerance differences. Accordingly, one value will always be slightly higher, regardless if the is a user interacting with the controller or not. Therefore, a threshold comparator was added to the system to sense only larger values from the user's presence and not respond to environmentally influenced reading when the user is idle.

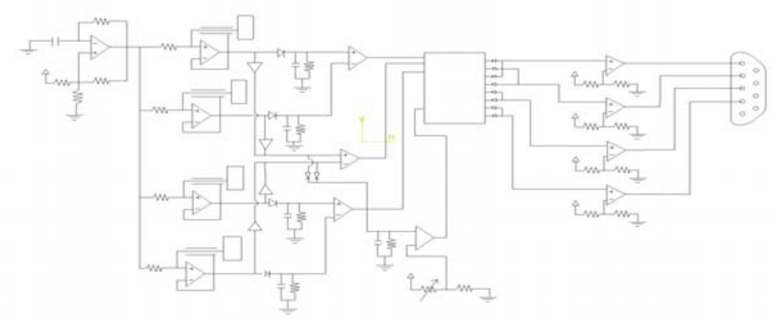


The circuit starts off with a single oscillator that drives all four capacitive plates. Using one single oscillator to drive all plates proved to be an important feature. When trying to compare values from four independent oscillators (of exact same values) for each capacitor, phase shifts from each prove to make comparisons very flaky (even with large RC values that flattened out the "shark-fin" performance of the low-pass capacitive plate sensors). Since capacitive interfaces are so sensitive to their surroundings, using a coaxial shielded wire was crucial. The shield was driven off following op-amp and ensured negligible interference. The next phase of the circuit compared the values of two channels in an op-amp, in this case left-v-right (comparator 1) and up-v-down (comparator 2). An additional third comparator in parallel compares which of the two groups is actually higher: left-&-right or up-&-down (comparator 3). These three logics are then fed into the three addresses of an eight channel demultiplexer, which adheres to the following logic to determine which output gates to deactivate (note: all outputs are steady state high unless activated low):

A0	A1	A2	O0	O1	O2	O3	O4	O5	O6	O7
0	0	0	0	1	1	1	1	1	1	1
1	0	0	1	0	1	1	1	1	1	1
0	1	0	1	1	0	1	1	1	1	1
1	1	0	1	1	1	0	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1
1	0	1	1	1	1	1	1	0	1	1
0	1	1	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	0
			L	R	L	R	U	U	D	D

The eight outputs were paired with diodes so that when one of two conditions where a particular direction is activated (ex: “left” is highest of the four and “up” is higher than “down” …or when left is highest of the four but “down” is higher than “up”) that output will be pulled low – thus activating that direction on the video game console. As mentioned before, since the values are not perfect and the capacitive plates will slightly interact with their environment, one value will always be high when the user is not interacting with the controller. Therefore a tunable comparator drives the enabler switch of the demultiplexer ensuring that the control only outputs a response to the game within a certain capacitive threshold from the user. This ensures no default direction or straying of direction when not specified by the user. Finding the proper threshold was a balancing act since it made the controller less sensitive. Accordingly, directional accuracy and triggering were inversely proportional to range. Lastly, since the video game console needs an input logic of 0 or 5 volts, pull-up comparators are used at the end since the presence of diodes prevented the combined four outputs to drive 5 volts.

To recap – a single oscillator drove four positioned low-pass capacitive sensors which were stretched with peak-detectors, compared, and addressed to a demultiplexer. When the user is close enough to surpass an adjustable threshold the multiplexer is then enabled to pull an according output low. These combined outputs are then pulled up and fed into the video game console’s controller input.

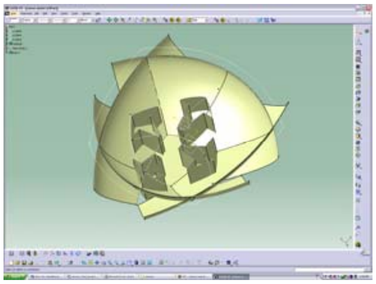


Initially the signals were slightly noisy and sluggish at a number of points along the circuit. Adding a number of electrolytic capacitors between the high and low power rails and ceramic capacitors between the op-amps’ Vcc and GND helped to improve the circuit’s overall performance. I believe (time permitting) cleaning up many of the long and loose wires on the breadboard and ensuring that each component and wire length of the low-pass capacitive sensor section are identical may have helped tightened the tolerances between the four inputs.

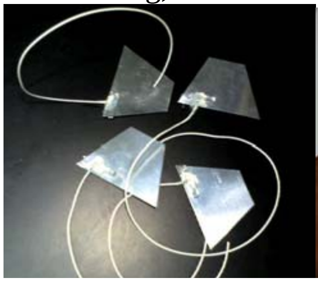
Control Interface Design

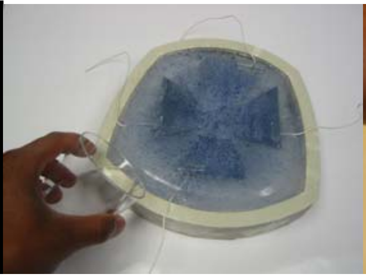
Multiple control interfaces were made by immersing thin aluminum plates into both urethane and rubber based materials.

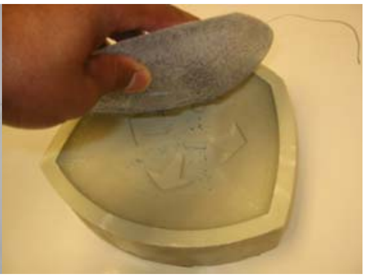
- First, a CAD model of the mold was created
- The mold was 3D printed on the z-corp, reinforced with resin and prepped with a mold-release solution

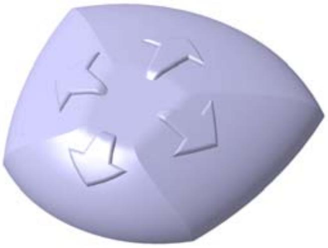


- Aluminum was cut to identical tapered plates and affixed with wires
- The urethane was poured into the mold and the plates were placed in position
- After curing, the control was removed connected to the circuit







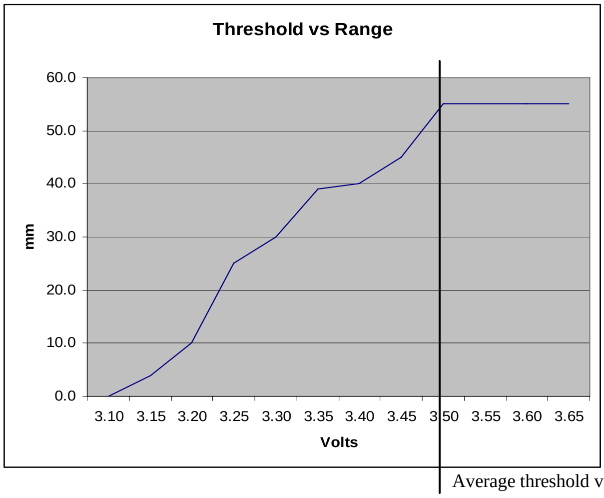


The aluminum plates were angled away from each other to match the pitch and roll user's hand position. The plates' geometries were also tapered to help increase the capacitive strength when the hand is positioned further in a particular direction.

Measurements / Results

Threshold v Range

Volts	mm
5.00	55.0
4.50	55.0
4.00	55.0
3.50	55.0
3.45	45.0
3.40	40.0
3.35	39.0
3.30	30.0
3.25	25.0
3.20	10.0
3.15	4.0
3.10	0.1



Average threshold voltage needed to keep control mute

As illustrated before, a threshold comparator was used to drive the enabling port of the demultiplexer. Although it helped to negate a faulty output when the control was not in use, it did decrease the sensitivity. As the above graph shows, as the threshold is widened the range of the sensor is increased. At a certain point (around 3.5 volts) the comparator is ineffective and the output is always on.

Range vs. Surface Area

SurfArea	Dist (mm)
160	60
250	75
500	100

position accuracy

dist (mm)	right bias angle (degrees)	down bias angle (degrees)
1	1	3
10	3	10
20	30	45
30	60	70

Range v Game play

dist (mm)	game level	error per level	costly errors
10	1.8	5.6	3
5	3.4	3.4	3
touch	4.6	2.6	2.1
pointing	4.1	3.1	2.5

A number of other measurements were taken to confirm that the capacitive sensor control was behaving properly. As expected, the range was increased with the larger controls which had larger aluminum plates. The initial left-v-right and up-v-down stages had consistent biases. The right and down plates of the sensor proved to be stronger and the user's hand needed to be placed a significant number of degrees opposite to active the direction. Lastly, game play was tested. As expected, I and others were able to surpass more levels when placing ours hands closer to the controller. The game was fairly sensitive to direction transitions and unfortunately triggered unwanted directions for a fraction of a second.

Conclusion

Overall the capacitive sensor worked fairly well considering some of the crude materials used. It consistently sensed hand position when a couple centimeters away. Adding some lag to the system may have improved the performance when in use for the video game since position transitions triggered some faulty readings. Eventually (less than a half second) the proper readings settled out; however, the split-second misdirection was enough to disrupt the game play. Also, with more effort I believe I may have been able to balance out the steady state readings of the compared readings by shortening up many of the wires and cleaning up the board. Having an adjustable threshold definitely helped adjust the sensor's sensitivity to the various environments. This was a good exercise in analog circuitry; nevertheless, I do believe I would have been able to build a significantly more accurate capacitive sensor using digital micro-controlled components.