Baran Kayhan
Report

The Musical Baby Doll

Renato came up with the musical baby doll idea when we were brainstorming about possible portable instruments the player could manipulate by interacting with it.

The doll incorporates many sensors to give the player a lot of control over the music it generates. The baby is from the Zapf Creation Company, and is called the Chou Chou doll. Unfortunately the baby was smaller than we hoped, and had a very soft body which spoiled some of our plans.
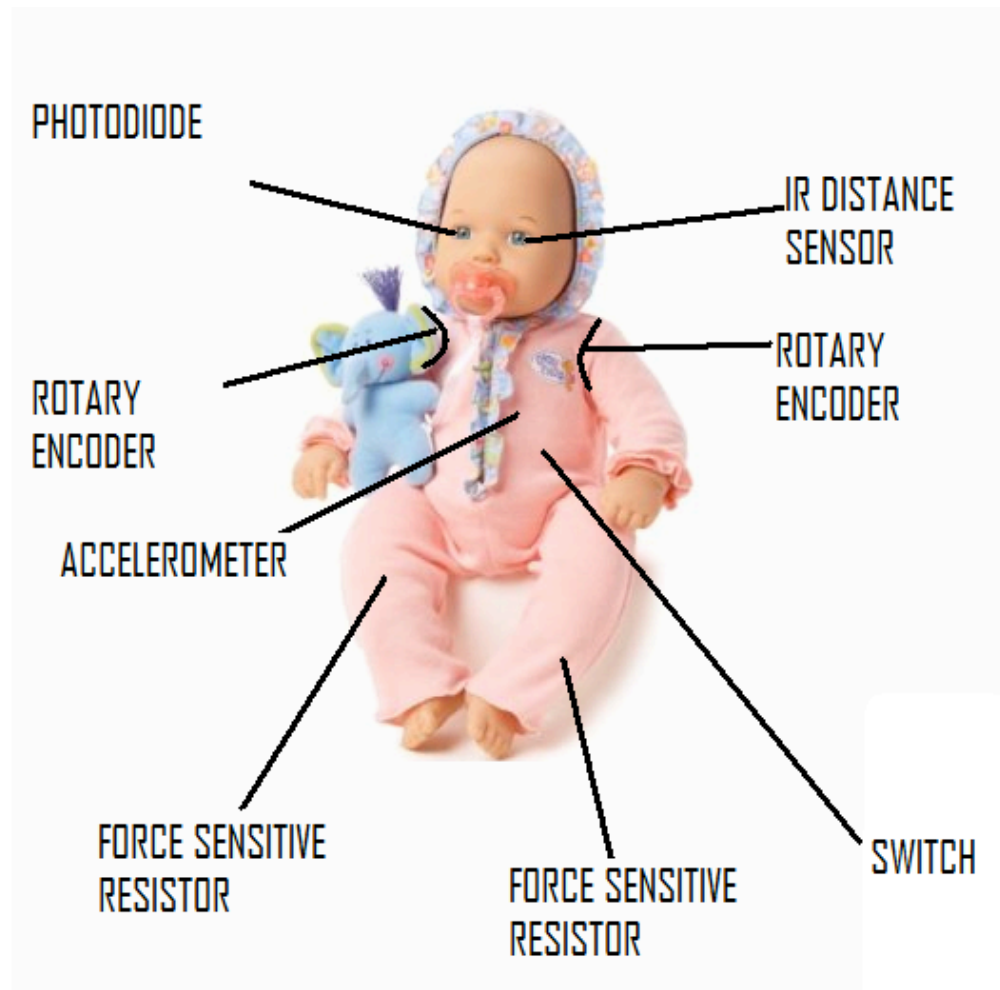
In addition to the sensors above, the baby had a Miditron unit attached to the back to transmit data wirelessly. This will be discussed in detail later on.

The baby, when turned on starts with a lullaby-ish music and the sensors attached to the body manipulate the sounds. You could also call the lullaby theme creepy if you have watched enough horror films to know that clowns and cute baby dolls can also be murderers.

Starting from the top and moving clockwise, the photodiode in the left eye was a 28.00$ Blue-light enhanced Silicon Detector from Edmund Optics. The basic operating procedure of a photodiode is when you expose it to light, electrons start moving around in the silicon, and the photodiode outputs some current.

Figure 2: Silicon detectors from Edmund optics

This current is usually very small and in order to convert that into a voltage we usually use a transimpedance amplifier, or a resistor. For quality output, I decided to make a transimpedance amplifier from a basic opamp. This is usually not very efficient for high speed and high precision applications (because of the inherent capacitance and the extra resistance in a basic opamp) but our project is neither:
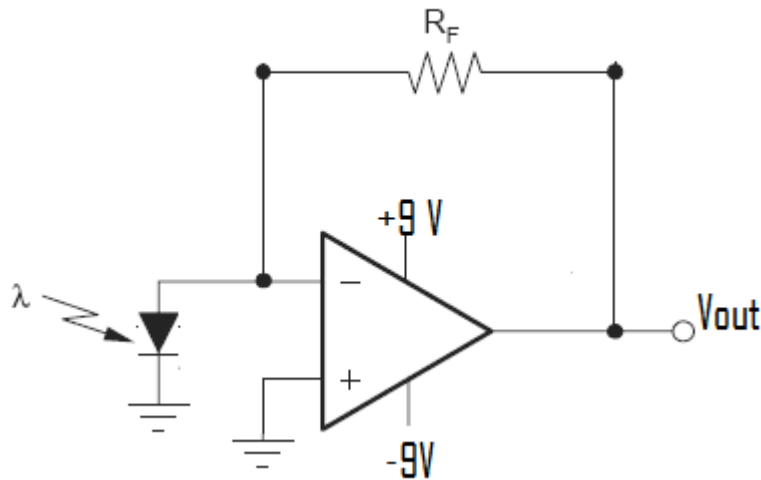
Figure 3: Current to voltage conversion and amplification with the opamp

I cut a small square breadboard piece with the Dremel tool and constructed the circuit. I tried a variety of resistors for RF to get good readings from the miditron and got one around 500 Ohms. The miditron also required that sensor be connected to the inputs in parallel with resistors, but when I did that with the photodiode, the ambient light reading was 0. We wanted the baby to be able to distinguish between 3 different light settings:

1- Dark
2- Room Light
3- Flashlight in the eye

The baby would make sounds accordingly to the light levels. In the dark, it would make a high pitched pipe sound to emulate crying, and would be soothed if the user is near the face. This feat is accomplished by putting an IR proximity sensor in the other eye.

Andrew connected the rotary encoders connected to the arms of the baby. He and Renato already had made a Reason patch and Renato decided to have the arms switch between patterns of notes they predetermined. This worked well.

For the belly switch, the team initially wanted to use an FSR but the stuffing was too soft inside the baby. Therefore we went with a switch that controlled the tempo of the theme music to simulate giggling. This switch assisted the 1G accelerometer inside the baby which mainly controlled the theme and arranged the music according to the tilt when being rocked.

The FSRs controlled chords that would be played when the baby's feet were "tickled." The left leg for exampled, played the D-minor chord. Here is the Maxpatch explaining every bit of the code

IR Sensor

Photodiode
making a
distressed sound
when light is <20

ctlin a 10

▷0   < 20

select 1

62   66   69   59

makenote 127 8000

noteout a 5

ctlin a 13

* 2.25

▷31

ctlout a 14 15

Accelerometer,
sensing the tilt at 70

ctlin a 14

▷0

select 70

timer

▷0

/ 24

metro 100

248

midiout a

notein a

select 64   select 127

1   1

&&

48

pipe 1

pipe 1

24

select 64   select 0

1   1

&&

pipe 1

pipe 1

The belly switch, when turned on
adjusts the tempo divider which is
controlled by the accelerometer.
When the switch is released the
tempo divider goes back to /24

Rotary encoders switch
Lullaby patterns in reason

notein a

select 60 61 62 63

0   1   2

ctlout a 3 1

ctlout a 3 2

ctlout a 3 3

ctlin a 11

> 90

pipe 3000

62   66   69   73

set 0, 127 2000   pipe 2000

set 127, 0 2000

line 0 47

line 127 47

makenote 127 8000

noteout a 6

ctlout a 1 6

The FSRs controlling a reason controller and
also playing the designated notes

ctlin a 12

> 90

pipe 3000

62   66   67   71

set 0, 127 2000   pipe 2000

set 127, 0 2000

line 0 47

line 127 47

makenote 127 8000

noteout a 6

ctlout a 1 6

My initial responsibilities were on the hardware side with Kyle, but our schedules did not match very well and he wanted to get all the wiring done after we started together. I designed and built the opamp circuit for the eye using an opamp from the EE department (a Motorola 7404 I think.), and Kyle wired the rest of the sensors to the Miditron.

Renato and Andrew had already made a lot of progress with the Reason patch. Renato and I spent the night and the morning in the lab later on to complete the Max Patch. Renato did most of the musical parts and I helped where I could with the programming. We ran into a lot of problems with the sensors because the wiring was messy and the exposed wires caused cross-interference and sometimes even shorts because the ground wire was also exposed and in very close proximity to the +5V wire. The miditron would suddenly stop working, and besides changing the battery, and resetting the hardware, I had to re-do some of the wiring and use electrical tape on the resistors and exposed wire-balls (the power line on the miditron had a wire coming out, which was in turn connected to a ball of metal, which was the soldered-together power lines from the other sensors). The fact that the sensors needed to be connected in parallel with resistors caused a lot of crowding and wiring faults. This would not be a problem if the baby was stationary, but constant movement and rocking quickly caused disconnections and shorts. Initially these problems were very frustrating because we did not know what caused it.

Another problem we ran into was the constant crashing of MAX, as evidenced in the demonstration. The jittery controllers turned out to be bad for timing a sequence.

-end-