Jason Chang, Taiyuan Hu
ECE110 Honors

Introduction

Problem Description

There are many studies involving posture, especially when sleeping or sitting. Inspired by the adjustable mattresses by companies like Sleep Number, we wanted to create a smart chair that would automatically adjust to your weight and position. Our project is the proof of concept and model of a chair that responds to inputs from three sources and moves the various parts accordingly.

Design Concept

In our design, we have three features to make this chair "smart": two sets of motors to adjust the height and the angle of the chair, and one bluetooth module. Two pressure sensors, one on the base and the other one on the back, feed their signal to an Arduino which would adjust the chair accordingly. One stepper motor is used to adjust the angle between the back and base. We glued the motor to the base and attached the axis to the back. The more pressure exerted on the sensor on the back of the chair, the more the chair would lean back. The reading from the base pressure resistance would cause the Arduino to control the movement of a servo motor under the base which would adjust the height of the chair. For the bluetooth module, we designed it to detect the user's phone, allowing them to manually control their desired height and reclining angle.

Analysis of Components

Sensor Characterization

To measure the weight of a person sitting or leaning back in the chair, we used two FSR406 from Interlink Electronics, which can detect as little as 100g and as much as 10kg of weight. These specifications, alongside their size, allowed us to easily test our project. Each force sensitive resistor has a resistance inversely proportional to the force exerted; as we had no controlled weights to characterize the sensors with, we instead utilized varying levels of force by pinching the sensor between our fingers and reading in the analog inputs with an Arduino.

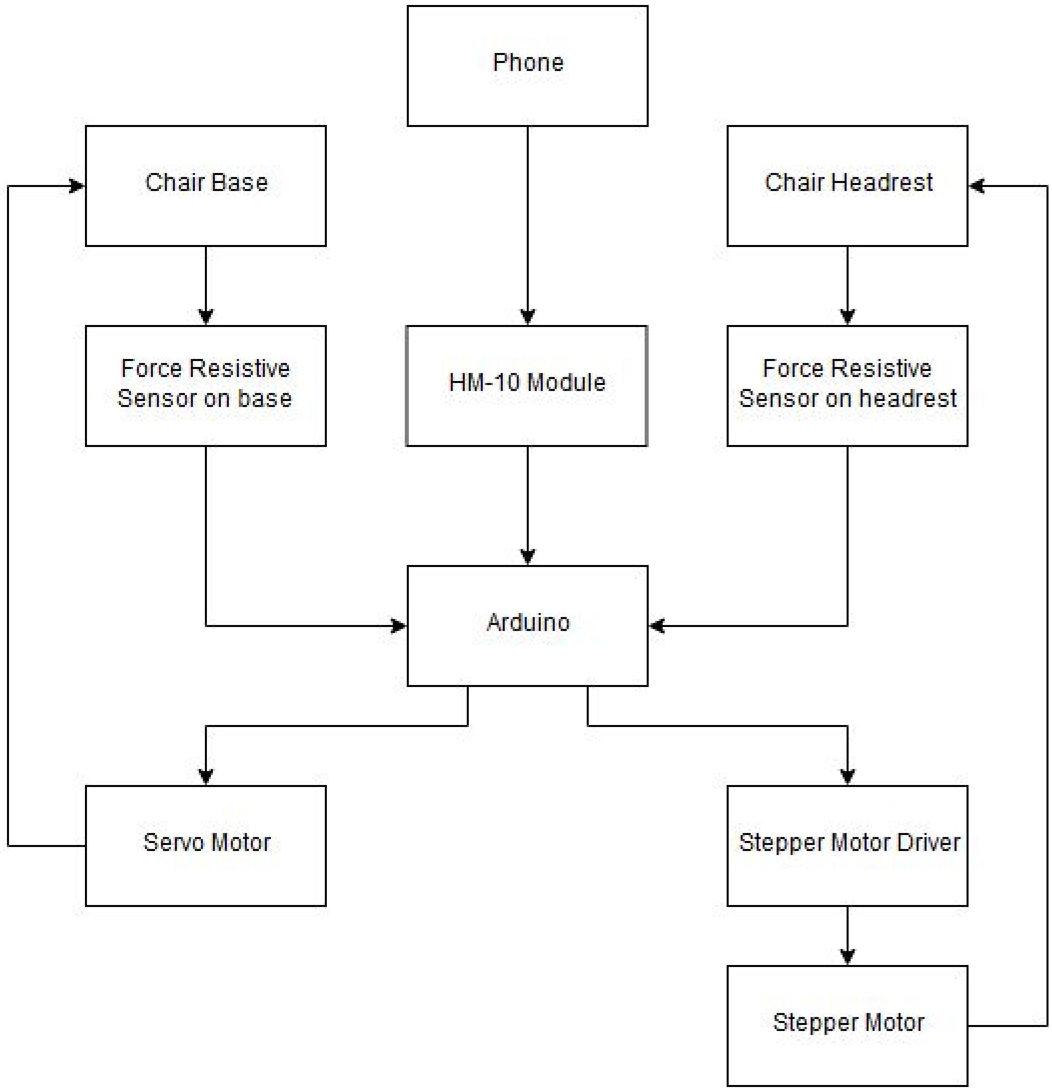
Force used	Arduino input
Holding the FSR in place	900
Medium pinch	800
Hard pinch	200

For the purposes of this project, we did not need a more accurate characterization because we also did not have any data on body types, and the purpose was to create a prototype that would respond to more or less force. Occasionally, the sensors would not respond unless they were slightly bent.

Design Considerations
Because the sensors were not extremely sensitive, they could not be placed directly on the model, as they would not register force properly; they were simply left connected to a breadboard instead. On a real product, more sensitive and accurate sensors would be required.

Design Description

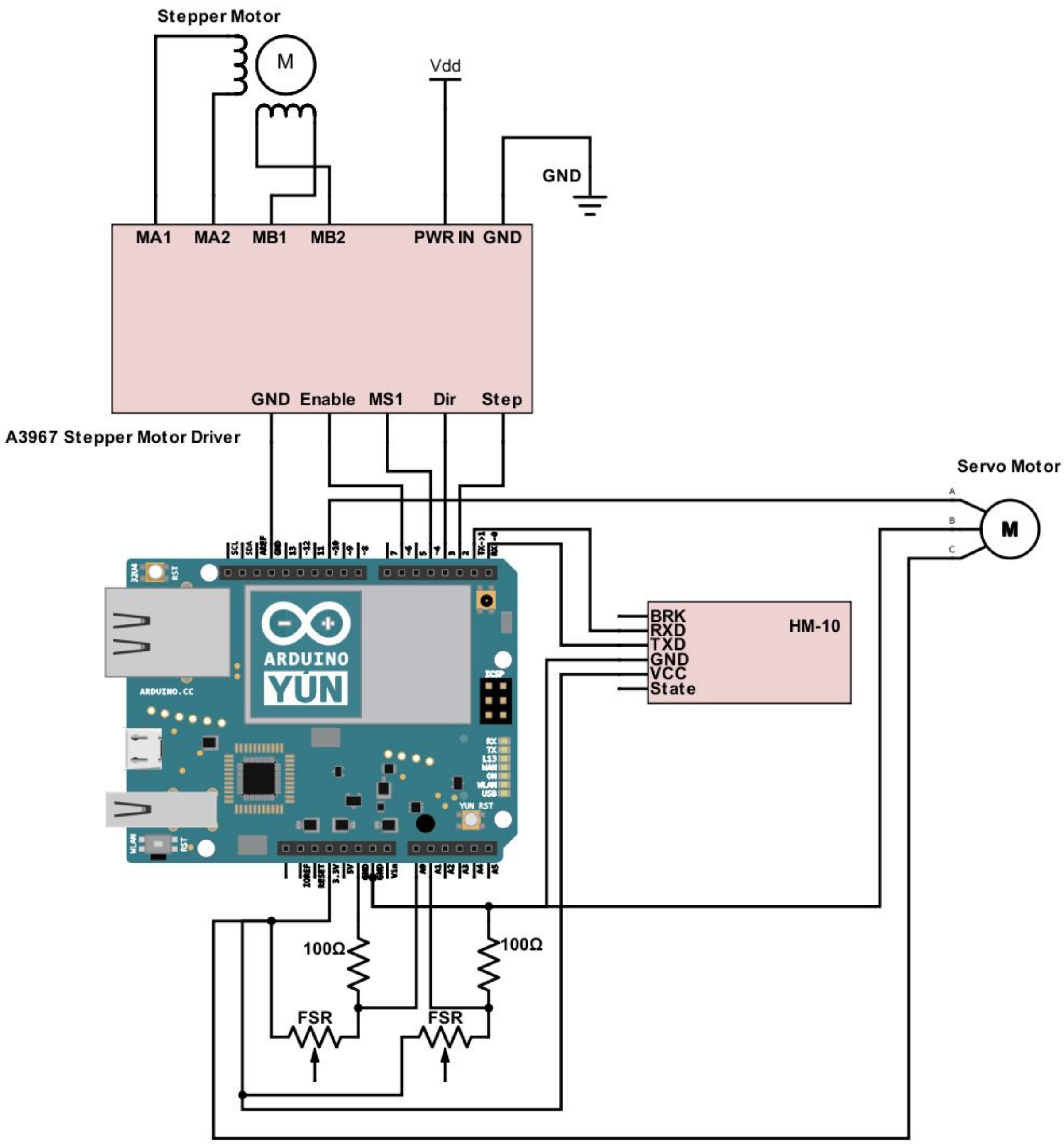
Block Diagram



Each part of the chair is controlled by a motor; a stepper motor for the back/headrest of the chair, to allow accurate adjustments to the angle, and a servo motor for the base, to raise and lower the chair. A force resistive sensor assigned to each part detects the force on the chair, and reports a signal (0 to 1023) inversely proportional to the force. The Arduino then compares the signals coming in with the previous signal, and tells either motor to move a certain number of steps based on the difference between the current signal and the previous signal. The stepper motor is also regulated by a driver, which allows easier PWM control and the ability to power the motor externally, as the Arduino cannot support the rated voltage (12V). The servo motor only requires 4.8V and as such can be powered directly by the Arduino. We also used the RemoteXY app and

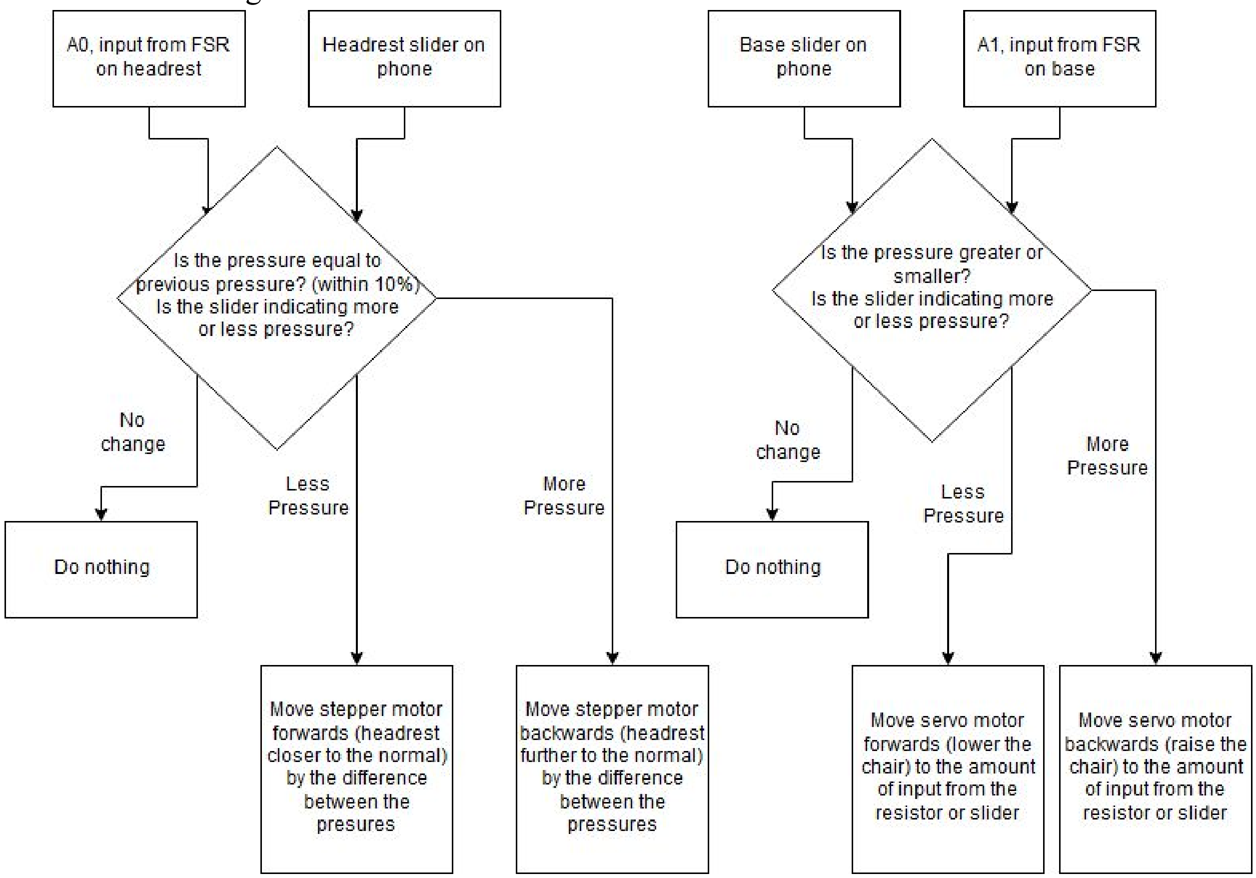
Arduino library to provide a simple UI on a phone app to control the motors; two sliders control the direction and magnitude of the motors.

Circuit Schematic



The arduino reads the data provided by two force resistive sensor through analog pin. After processing them, arduino sends one signal to EasyDriver Stepper Motor Driver which controls the stepper motor to adjust the angle between base and back, the other signal to servo motor. Scaling the signal input of A1 to (0, 1023), the arduino converts it to output signal and drive the servo motor from 0 to 90 degrees according to this signal. The greater the force exerted on the force resistive sensor, the bigger the data input, the more servo motor rotates, and thus, the higher the chair would raise. For the bluetooth module, once it senses the signal from a paired device, it communicates using the RemoteXY app and library, which then sends a signal to the Arduino to be processed and turned into motor commands.

Arduino Block Diagram



The Arduino performs a comparison for each motor; for the stepper motor, it reads in the input from the sensor on the headrest and checks it against a variable currentPressure, which defaults to zero. If the pressures are within 10% of each other, the Arduino does nothing; otherwise, it tells the stepper motor to move forwards or backwards according to the relative difference between the two pressures, and it replaces the old currentPressure with the new input. For the servo motor, the Arduino reads in the input from the sensor on the base and adjusts the motor accordingly; rather than comparing it to a variable, each value of the input corresponds to a different servo location. This was easier to implement with the servo library.

Conclusions

The circuit we designed worked; both motors can rotate according to the reading from the pressure sensor. The biggest issue of the whole project is attaching the motor to the 3D printing parts. Even though two parts could be matched according to the diagram on the inventor, the actual printed parts are not as accurate as the blueprint due to tolerances being high. After several iterations of the chair base, we finally printed out two matched parts. Yet fixing two engines to these parts is still difficult for us. Using two different kinds of glue, we fixed the stepper motor, which is responsible for changing the angle between back and base, to the 3D printed parts. However, the servo motor which is designed to change the height of the base cannot be fixed to the base. One of the problems we had with the servo motor was changing the rotational movement to translational movement. In addition, our bluetooth component was non-functional; despite achieving communication between the serial monitor and the bluetooth module, unfortunately the phone app crashed every time we tried to connect. However, we both learned a lot about putting a project together that combined logic, sensors, and mechanical aspects.