Penny Robot

Richard grimmett | 4/3/2025

ECEN 461: advance embeded systems

Kat lucido & Rylan Jurgens

2025

# Objectives

* Create a budget friendly desk toy
* Test different sensors and pair with existing drivers and libraries
* Develop tests and use cases to decide upon the best final components

# Procedure

## Overview

To design the project, we first broke the idea into different parts. We decided we wanted the main components to be a “puppet”, a main box or compartment, a motor, and a sensor. The puppet would be on top of the box and have its hands attached to a guide stick. This guide would then be attached to the motor. If the motor spins it will cause the puppet to rock back and forth. This rocking would be the key feature of the product.

To turn the motor on or off, we wanted it to somehow be able to detect coins, or other objects placed on the box. The more objects the faster the motor would spin.

And finally, we would have a cheap Arduino provide the processing power to facilitate these functions.

The result would be a cheap and easy toy to have on a desk and provide some entertainment or learning activities through the increase in puppet shaking directly proportional to the objects placed on the box.

## Research

With the desired parts outlined we then had to find parts that could fit into our design. We decided that the Arduino Uno would be a good fit, as it is readily available and low-cost so fits the design of this being a desk toy. The motor and object sensor were not so easy to decide on. Kat decided to lead the research the different motors while Rylan went with the Object Sensor.

### Motor

From talking with our professor, and personal research we found three different types of motors that we would investigate: Servos, Stepper Motors, and DC Motors.

We found that Servos were usually a bit more expensive and offered significantly more utility and precision. While DC Motors were the cheapest and had the least precision. Stepper Motors were in between.

First, we tested a Stepper Motor that we had lying around in some old kits. A few libraries were found for Arduino that offered better control, but the RPM was limited. After a week of testing and looking at potential replacements we decided that DC motors were our best bet for getting a decent RPM to shake the doll.

The new DC Motor had to be supplemented with a L29\*N Motor Driver to better control the RPM with our code on the Arduino.

### Object Sensor

For detecting the objects, we looked at a few different types of sensors. Weight, InfraRed, Touch, and AI. The InfraRed and Touch were set aside after we decided it would be difficult to tell multiple objects apart from just one. While AI would pose a problem with the processing power of the Arduino Uno.

The weight sensor then became our focus as we started looking at the different types of sensors that can measure weight. Our search led us to looking at Load Cells and the HX711 amplifier. After some testing and deciding on a good library we were ready to move onto putting them all together.

Later on, though, we found problems with the Load Cells we bought as they were prone to drift if the calibration was inaccurate and lacked sensitivity enough to detect little weight changes. We therefore made the decision to turn around and begin to use a vibration sensor instead. Within our design limitations, the vibration sensor let us more precisely identify when objects were placed on the box.

|  |  |  |
| --- | --- | --- |
| Part Name | Model | Quantity |
| Lafvin Uno | ATMEGA16U2 | 1 |
| DC motor driver module | L298N | 1 |
| Mini motors | - | 1 |
| Vibration sensor | SW-420 | 1 |
| Digital load sensor | - | 1 |
| ADC module | HX711 | 1 |
| 5V stepper motor | 28BYJ-48 | 1 |
| Driver board | ULN2003 | 1 |
| Custom 3D Print | - | 5 |

### Code

const int vibrationPin = 2;    // Connect to the sensor's DO pin

const int in1 = 8;             // L298N IN1

const int in2 = 9;             // L298N IN2

const int enablePin = 10;      // L298N ENA (PWM for speed)

void setup() {

  pinMode(vibrationPin, INPUT);

  pinMode(in1, OUTPUT);

  pinMode(in2, OUTPUT);

  pinMode(enablePin, OUTPUT);

  // Start with motor off

  digitalWrite(in1, LOW);

  digitalWrite(in2, LOW);

  analogWrite(enablePin, 0);

  Serial.begin(9600);

}

void loop() {

  int vibrationDetected = digitalRead(vibrationPin);

  if (vibrationDetected == HIGH) {

    Serial.println("Vibration detected! Fan ON.");

    digitalWrite(in1, HIGH);

    digitalWrite(in2, LOW);

    analogWrite(enablePin, 500);  // adjust the speed (0–255)

    //delay(90);  // Let it spin up

    //analogWrite(enablePin, 250);  // Then slow down

  } else {

    Serial.println("No vibration. Fan OFF.");

    digitalWrite(in1, LOW);

    digitalWrite(in2, LOW);

    analogWrite(enablePin, 0);

  }

  delay(100);  // Small debounce delay

}

# Results

A toy figurine holding a stick and a stick on a green box with coins and a computer monitor

AI-generated content may be incorrect.

The Original plan was to find a wooden box that would fit out components. But we decided to try and learn some more skill and investigate 3D printing our own box and some adapters to fix the motor, stick, and puppet together.

After looking into different CAD software and realizing our time constraints as we neared the end of the semester and other projects and classes, we decided to ask a friend that had some previous experience to come and help us develop the parts. Working with him we were able to get a basic box printed and the motor, sensor, and doll connected, and controller by the Arduino.

Some Errors are still present as we found that Load Cells, we got were not the most sensitive to small changes in the grams and were prone to drift if the calibration was off. The motor also has some small defects as we went with an affordable version that requires larger steps in RPMs than we wanted to have originally.

# Conclusion

This project allowed us to grow our understanding of the various types of sensors and motors available on the market. We spent a considerable amount of time testing various types and researching the alternatives that we now feel confident in applying the different options to a larger variety of projects.

We also feel much more ready to create tests, and to be able to narrow down the cause of various errors in our wiring and programming.

The final and most important aspect we learned from this was how to build a project from the ground up. We had to do all the groundwork to decide on the type of sensor, to the type of motor, power management, and code the Arduino. This allowed us to gain a deeper understanding of the whole design process rather than just how to put something together from a kit.