***University of Waterloo***

*Department of Mechanical and*

*Mechatronics Engineering*

# Coin Thickness Measurement Device

Due: March 27, 2017

Prepared by: Cameron Blair, Eric Tweedle & Morgan Yeung

Prepared for: Peter Teertstra, MTE 201

# 1.0 Design Summary

The purpose of this project was to design and build a measurement apparatus that measures either water temperature, air pressure, or the thickness of the coin. Upon consideration, it was decided that the device built should measure distance. The distance that was measured was the thickness of different Canadian coins minted from 2013 to present.

## 1.1 Theory of operation

The device is made up of a servo motor that rotates an arm, and a capacitive touch sensor to detect when the arm touches the coin. The Arduino controlling the servo motor keeps track of the angle that the arm has travelled. Using this recorded angle and the length of the arm, the change in height of the end of the arm can be calculated. Since the initial height is known, the thickness of the coin can be determined.

The layout of the apparatus is shown in Figure 1. In the diagram, h1 is the thickness of the coin, h0 is the initial height of the arm, Δh is the distance travelled by the arm before it makes contact with the coin, L is the length of the arm, and θ is the angle that the arm travels. It is seen that

$$h_1 = h_0 - \Delta h \quad (1)$$

Using trigonometry, it can be shown that

$$\Delta h = L \sin(\theta) \quad (2)$$

And the thickness of the coin defined using only known values is

$$h_1 = h_0 - L \sin(\theta) \quad (3)$$

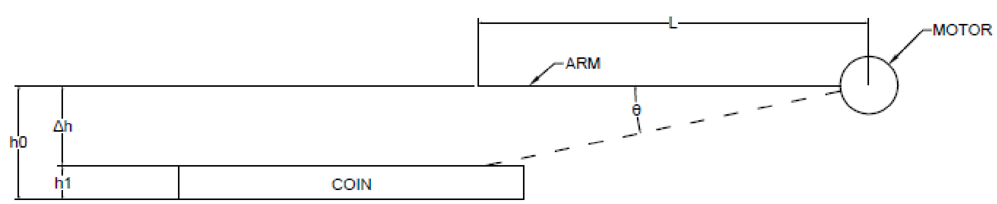



**Figure 1:** Drawing of layout

This design is based off the assumption that the change in angle will be small enough that the corresponding change in height of the arm can be approximated as linear.

The capacitive touch sensor detects the contact of the arm with the coin by comparing the time it takes for two connected Arduino pins to have the same state. When something conductive, in this case the wire on the arm connected to the can, touches the coin, it takes longer for the receiving pin to have the same state as the sending pin [1].

The circuit used for the measurement device can be found in Appendix B. The circuit can be broken down into four smaller independent circuits that work together with the Arduino Uno to make the final device. The four circuits are for the LCD display, the pushbutton, the capacitive sensor and the servo motor. Inspiration for the circuits was gathered from an Arduino projects book [1]. The schematic for the circuit was designed using the Fritzing Project [2].

## 1.2 Measurement system elements

This device has all the common elements of a measurement device, namely, a sensor, a signal modification system, and an indicator. The sensor has two components: the servo motor to measure the angle that the arm travels, and the capacitive touch sensor to detect the coin. The signal modification is a program on the Arduino that converts the angle to a thickness based on the derived function. Finally, the indicator is an LCD display which outputs the thickness of the coin.

# 2.0 Calibration

The calibration of the device used coins of various denominations and a digital caliper. The caliper had an accuracy of 0.001" or 0.02mm [3]. Three coins of each denomination ranging from nickels to toonies were used. After measuring the coins using the caliper, the coins were measured using the device. For calibration purposes, the outputs were the angles, rather than the theoretical value of the coin's thickness. The results of each measurement are shown in Table 1. The measured angles were then plotted against the actual thickness of the coins. This is shown in Figure 2.

**Table 1:** Calibration measurements

| Coin | Thickness (mm) | Steps (°) |
|---|---|---|
| **Nickel 1** | 1.79 | 18 |
| **Dime 1** | 1.18 | 23 |
| **Quarter 1** | 1.6 | 20 |
| **Loonie 1** | 1.93 | 19 |
| **Toonie 1** | 1.69 | 20 |
| **Nickel 2** | 1.79 | 19 |
| **Dime 2** | 1.22 | 23 |
| **Quarter 2** | 1.63 | 21 |
| **Loonie 2** | 1.92 | 19 |
| **Tonie 2** | 1.79 | 20 |
| **Nickel 3** | 1.79 | 19 |
| **Dime 3** | 1.2 | 23 |
| **Quarter 3** | 1.63 | 21 |
| **Loonie 3** | 1.93 | 19 |

| | | |
|---|---|---|
| **Toonie 3** | 1.77 | 20 |

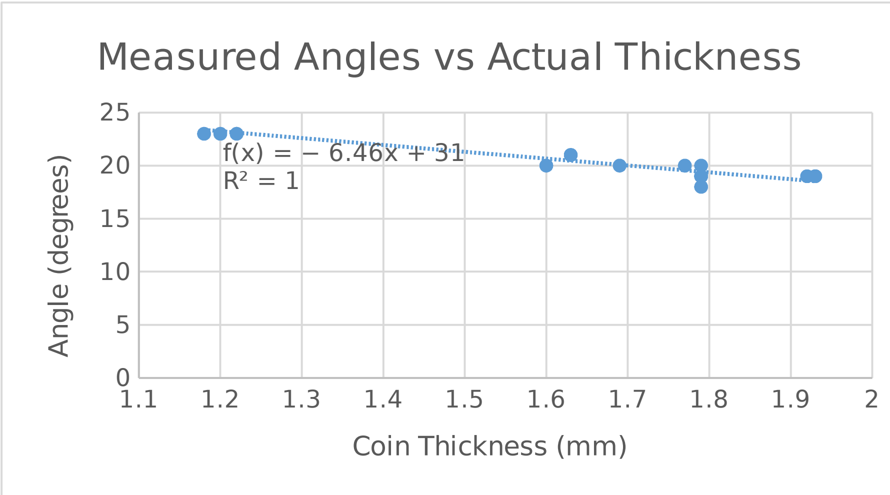



**Figure 2:** Calibration curve for the measurement device

The y-intercept was set to 31 because this is the angle of the arm that causes it to hit the base of the measurement surface. From the $R^2$ value, it is evident that this curve is a strong fit for the data.

The resulting function for the measured thickness is

$$h = \frac{-\theta - 31}{6.4633} \quad (4)$$

## 3.0 Uncertainty analysis

Once the function for the signal modification was determined through calibration, the measured values could be compared to the actual values. The results are shown in Table 2.

| **Table 2:** Deviation analysis of measured and actual valuesXCoin | **Width (mm)** | **Steps (°)** | **Value (mm)** | **Deviation (mm)** |
|---|---|---|---|---|
| **Nickel 1** | 1.79 | 18 | 2.011356 | -0.22136 |
| **Dime 1** | 1.18 | 23 | 1.237758 | -0.05776 |
| **Quarter 1** | 1.6 | 20 | 1.701917 | -0.10192 |
| **Loonie 1** | 1.93 | 19 | 1.856637 | 0.073363 |
| **Toonie 1** | 1.69 | 20 | 1.701917 | -0.01192 |
| **Nickel 2** | 1.79 | 19 | 1.856637 | -0.06664 |
| **Dime 2** | 1.22 | 23 | 1.237758 | -0.01776 |
| **Quarter 2** | 1.63 | 21 | 1.547197 | 0.082803 |

| | | | | |
|---|---|---|---|---|
| **Loonie 2** | 1.92 | 19 | 1.856637 | 0.063363 |
| **Toonie 2** | 1.79 | 20 | 1.701917 | 0.088083 |
| **Nickel 3** | 1.79 | 19 | 1.856637 | -0.06664 |
| **Dime 3** | 1.2 | 23 | 1.237758 | -0.03776 |
| **Quarter 3** | 1.63 | 21 | 1.547197 | 0.082803 |
| **Loonie 3** | 1.93 | 19 | 1.856637 | 0.073363 |
| **Toonie 3** | 1.77 | 20 | 1.701917 | 0.068083 |

For a clearer comparison, the deviation and actual thickness of the coins were plotted in Figure 3.

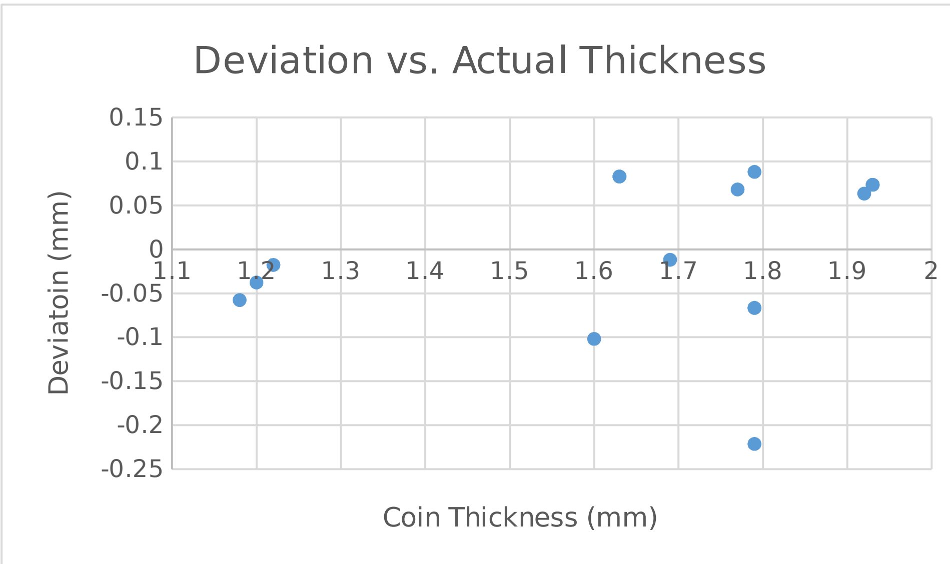



**Figure 3:** Deviation plot for the measurement device

Based on the plot and the graph, the largest deviation was -0.22136 mm. The device is just as likely to have measure the coin as larger than it is as smaller, the accuracy is ±0.222 mm (±0.009 in).

While it may not be apparent from looking at the deviation plot, there is in fact a systematic error. It stems from the fact that the servo motor can only rotate an integer number of degrees. When the coins are grouped by denomination, this becomes more apparent. This is shown in Figure 4. When comparing these results to the values in Table 2, notice that the major differences in the deviation for a specific denomination occur where there was a difference in the angle measured. Each degree corresponds to approximately 0.2 mm.

The obvious scatter in the data corresponds to the random error, which in general is due to the idiosyncrasies of the device in terms of when it registers that it has touched the coin. The error is also partly due to inconsistencies in positioning the coin before the measurement starts. Depending on which loop of the wire touches which part of the coin, there will be variations.

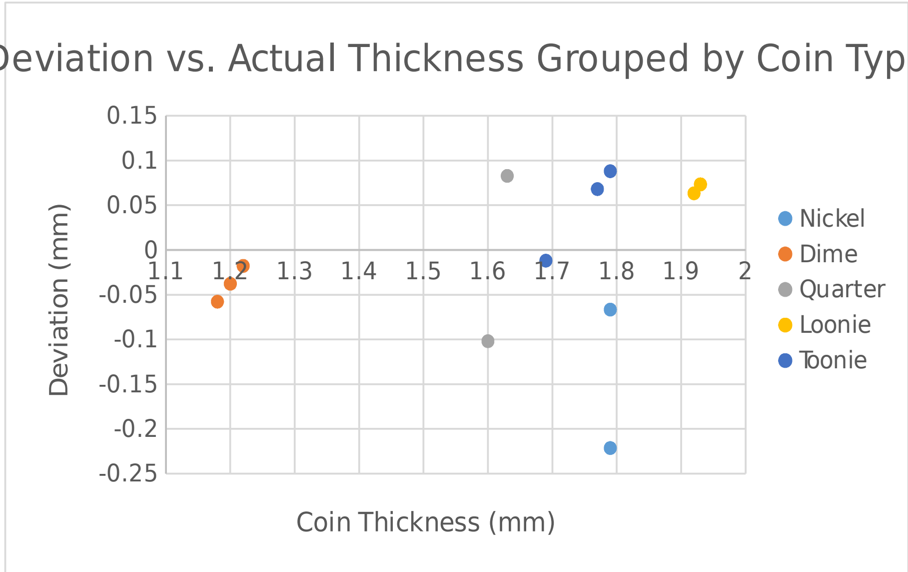



**Figure 4:** Deviation plot for the measurement device, grouped by coin denomination

## 4.0 Conclusion

The goal of this project was to design a measurement system to measure the thickness of a coin. The solution proposed was an Arduino controlled servo motor with an arm attached and a capacitive touch sensor to detect when the arm contacts the coin. Then through calibration the number of steps made by the servo motor was related to the thickness of the coin. This system has an accuracy of ±0.222 mm (±0.009 in).

There are two major ways in which this design could be improved. The first is to use a servo motor that does steps less than 1°. This allows more precise measurements of the angle, which will also allow for a more precise measurement of the thickness. The other is to have a way of positioning the coins more consistently, to reduce the random error due to the variation in the position.

-

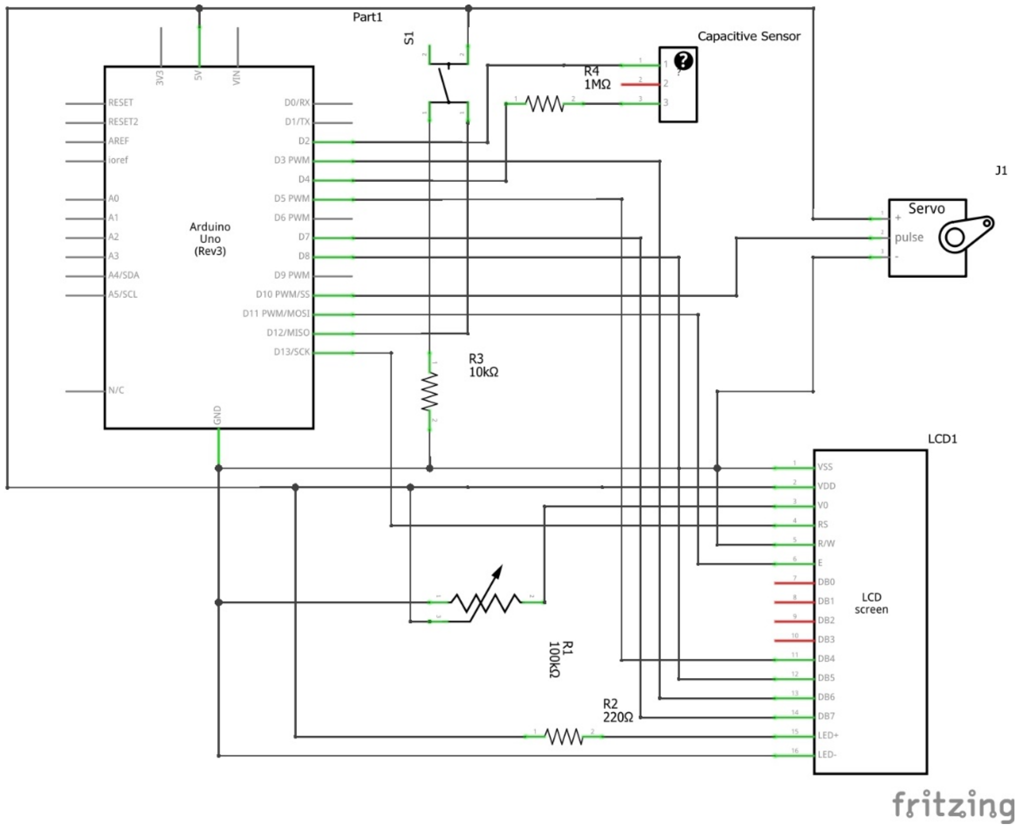
# Appendix A: References


[1] S. Fitzgerald and M. Shiloh, *Arduino Projects Book,* Third Issue. Torino, Italy: September 2012

[2] Fritzing Project. Friends-of-Fritzing foundation. Version 0.9.3b, Windows, June 2, 2016. Open Source. [Online]. Available: http://fritzing.org/home/ (March 25, 2017).

[3] Canadian Tire. (2017, January 6). Mastercraft Digital Caliper [Online]. Available: http://www.canadiantire.ca/en/pdp/mastercraft-digital-caliper-0586800p.html#Questions (March 26, 2017).

# Appendix B: Circuit Schematic




## Assembly List

| Label | Part Type | Properties |
|---|---|---|
| Capacitive Sensor | Tin Foil | true; pins 3; label ?; hole size 1.0mm,0.508mm; layout Single Row; pin spacing 100mil; package THT |
| J1 | Basic Servo | |
| LCD1 | LCD screen | type Character; pins 16 |
| Part1 | Arduino Uno (Rev3) | type Arduino UNO (Rev3) |
| R1 | Rotary Potentiometer (Small) | size Rotary - 9mm; track Linear; type Rotary Shaft Potentiometer; package THT; maximum resistance 100kΩ |
| R2 | 220Ω Resistor | resistance 220Ω; tolerance ±5%; bands 4; pin spacing 400 mil; package THT |
| R3 | 10kΩ Resistor | resistance 10kΩ; tolerance ±5%; bands 5; pin spacing 400 mil; package THT |
| R4 | 1MΩ Resistor | resistance 1MΩ; tolerance ±5%; bands 4; pin spacing 400 mil; package THT |
| S1 | Pushbutton | package [THT] |

# Appendix C: Source Code

```
/*

Source Code for coin thickness measuring system

*/

#include <Servo.h>

#include <CapacitiveSensor.h>

#include <LiquidCrystal.h>


CapacitiveSensor capSensor = CapacitiveSensor(4,2);

LiquidCrystal lcd(13,11,5,8,3,7);

Servo servoMain;


int threshold = 70;

int switchState = 0;

bool start = false;

const float initialPos = 90;

float pos = initialPos;

int steps = 0;


void setup() {

  servoMain.attach(10);

  lcd.begin(16,2);

  lcd.print("press button");

  pinMode(12,INPUT);

  pinMode(13,OUTPUT);

  Serial.begin(2400);


}
```

```
void loop() {
  long sensorValue = capSensor.capacitiveSensor(30);
  switchState = digitalRead(12);
  Serial.print(digitalRead(12));
  if (switchState == 1){
    steps = 0;
    start = true;
  }
  if(start){
    if (sensorValue > threshold){
      lcd.print(" Done");
      start = false;
    }

    servoMain.write(pos);
    pos -=1;
    steps += 1;
    if (pos == 0){
      pos = initialPos;
    }
    lcd.clear();
    lcd.print(steps);
    delay(1000);
  }
  delay (10);
}
```