Project Title: Arduino Vending Machine

Group Members: Anthony Guzman, Brian Montoya, Ernesto Torreblanca

1. Device Functionality

Our project is modeled after a conventional vending machine, the difference being we used an NFC card reader as our method of transaction. Using a card, we modeled “currency” in order to “buy” snacks. First, in order to know the cost of each snack, the buyer must push the buttons corresponding to the snack. This is labeled using different colors. Then from there, the buyer can scan the special card which the NFC reader will read and display the buyer’s balance. If the buyer has enough balance he/she can choose the snack they want and the machine will dispense it. If the card does not have enough money, then the LCD screen will display not enough balance and will not dispense an item. The act of dispensing occurs by the button commanding the specific servo, where the snack is, to turn the coils which will push the snack forward until it falls. This will successfully occur only when there is enough balance in the card. Finally, in order to actually add currency, we used a chip which can also be scanned on the NFC which will add a balance.

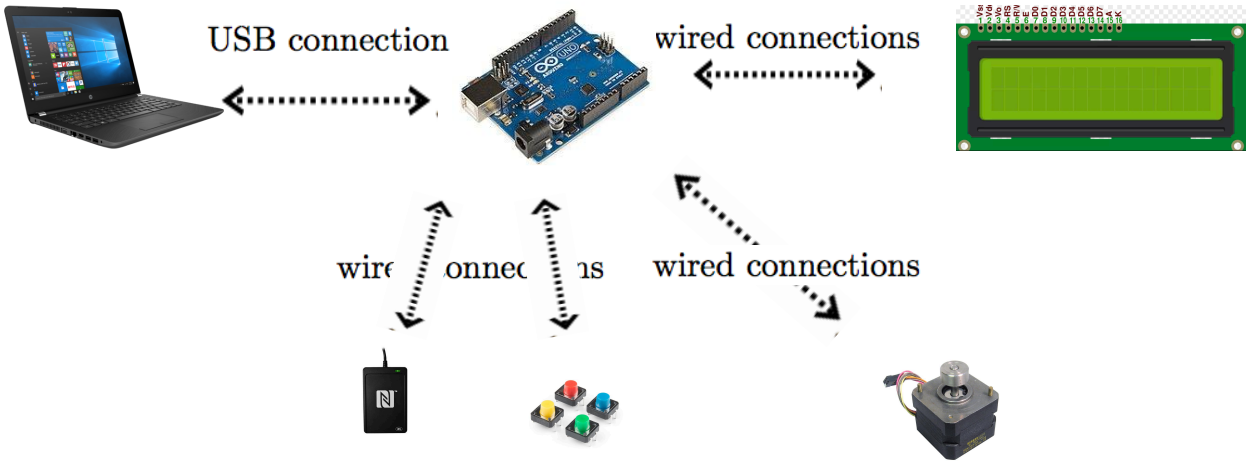


Figure 1: Outside of the Vending Machine (Instead of a coin acceptor, we will use a NFC key reader)

2. Hardware Components

- NFC Card/Chip reader [Source: Amazon($12)]
- 4 Arduino Push Buttons [Source: Amazon($7)]
- 4 Arduino Continuous Servo Motors [Source: Amazon($32)]
- LCD Screen Display
- Acrylic Screen [Source: Home Depot ($10)]
- Arduino Uno
- Aluminum Wire[Source: Home Depot($4)]
- Cardboard[Source: Home Depot($2)]

- Foam Board[Source: UCSB Bookstore($2)]
- 12V Battery Pack [Source: ECE Shop($8)]

3. Design Timeline

Week 5: During this lab session we had already acquired the components we needed, so we were playing around with them and testing that they worked. At this point we got the servos to respond to the buttons.

Week 6: During this lab session we were able to build the circuits for the LCD, buttons, and servos. We started to cut and build the cardboard into the vending machine, but not fully. We were able to get some programming so the LCD, buttons, and servos worked together.

Week 7: We got the NCF to work during this session when the card would be tapped. From here we were able to program all of the various components we had with the NFC. This meant we had to scan the card before we could press the button to rotate the servo motors.

Week 8: Since we had most of the programming figured out we focused more on the hardware by soldering the button onto a perfboard as well as making the coils needed. We then glued these coils onto the servos.

Week 9: We started to get the vending machine close to completely built by adding the foam board as well as getting all the servos situated and finishing up on the soldering. Minor changes were made to programming such as displaying prices and instructions before buying.

Week 10: We fully assembled the box by placing the Acrylic screen, making cuts for the LCD, buttons, and the NFC. We also got the placement of the arduino situated so that it could stay inside. We also figured out how we would be able to access the inside once the vending machine was complete. We also tested different snacks to make sure the machine dispensed. Finally we treated ourselves as customers and made sure everything went smoothly when trying the vending machine.

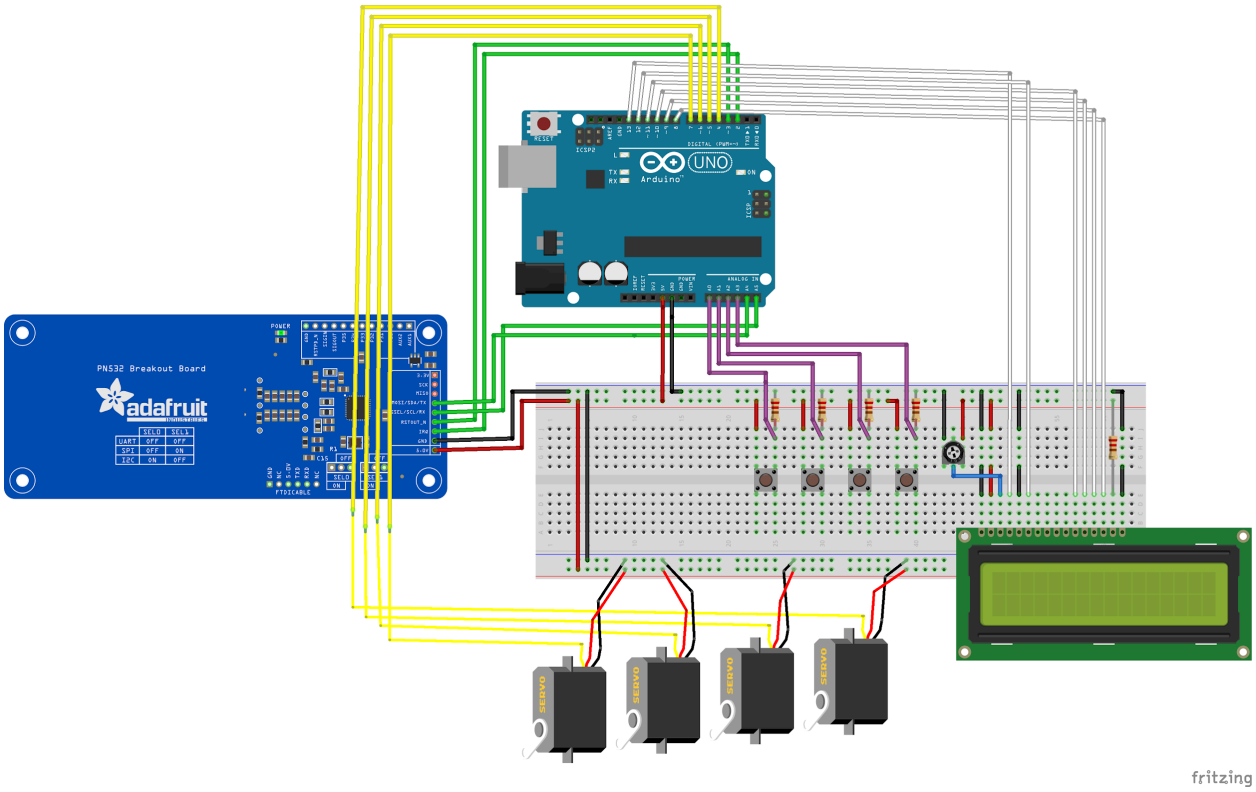
4. Software Design

```
#include <LiquidCrystal.h>
#include <Servo.h>
#include <Wire.h>
#include <SPI.h>
#include <Adafruit_PN532.h>
#define PN532_IRQ   (2)                                         // Several Libraries
#define PN532_RESET (3)
Adafruit_PN532 nfc(PN532_IRQ, PN532_RESET);
#if defined(ARDUINO_ARCH_SAMD)
   #define Serial SerialUSB
#endif

Servo myservo;                                                  // Sets up the Servo motor

const int rs = 13, en = 12, d4 = 11, d5 = 10, d6 = 9, d7 = 8;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);                      // Sets up the LCD Text with the Pins
char * Text = "Scan Card Now.              ";                   // Text "Scan Card Now"
int iLineNumber = 0;
int iCursor = 0;

int cardid = 0;                                                 // Variable for NFC Tag

float cash = 0;                                                 // Card Balance ($)
float Blueprice = 1.00;                                         // Prices for Blue Button
float Greenprice = 1.50;                                        // Prices for Green Button
float Yellowprice = 1.75;                                       // Prices for Yellow Button
float Redprice = 2.00;                                          // Prices for Red Button

unsigned long time_now = 0;

void setup()
{
  Serial.begin(9600);
  Serial.begin(115200);

  pinMode(14, INPUT);                                           // Setups the Buttons
  pinMode(15, INPUT);
  pinMode(16, INPUT);
  pinMode(17, INPUT);

  lcd.begin(16,2);                                              // Intializes the LCD Screen

  #ifndef ESP8266                                               // Intializes the NFC Reader
    while (!Serial);
  #endif

  nfc.begin();                                                  // Intializes the NFC Reader
  uint32_t versiondata = nfc.getFirmwareVersion();
  nfc.SAMConfig();                                              // Read RFID tags
}

void loop()
{
  if(millis() > time_now + 500){
       time_now = millis();                                     // Updates the LCD screen every 0.5 second, until other actions
```

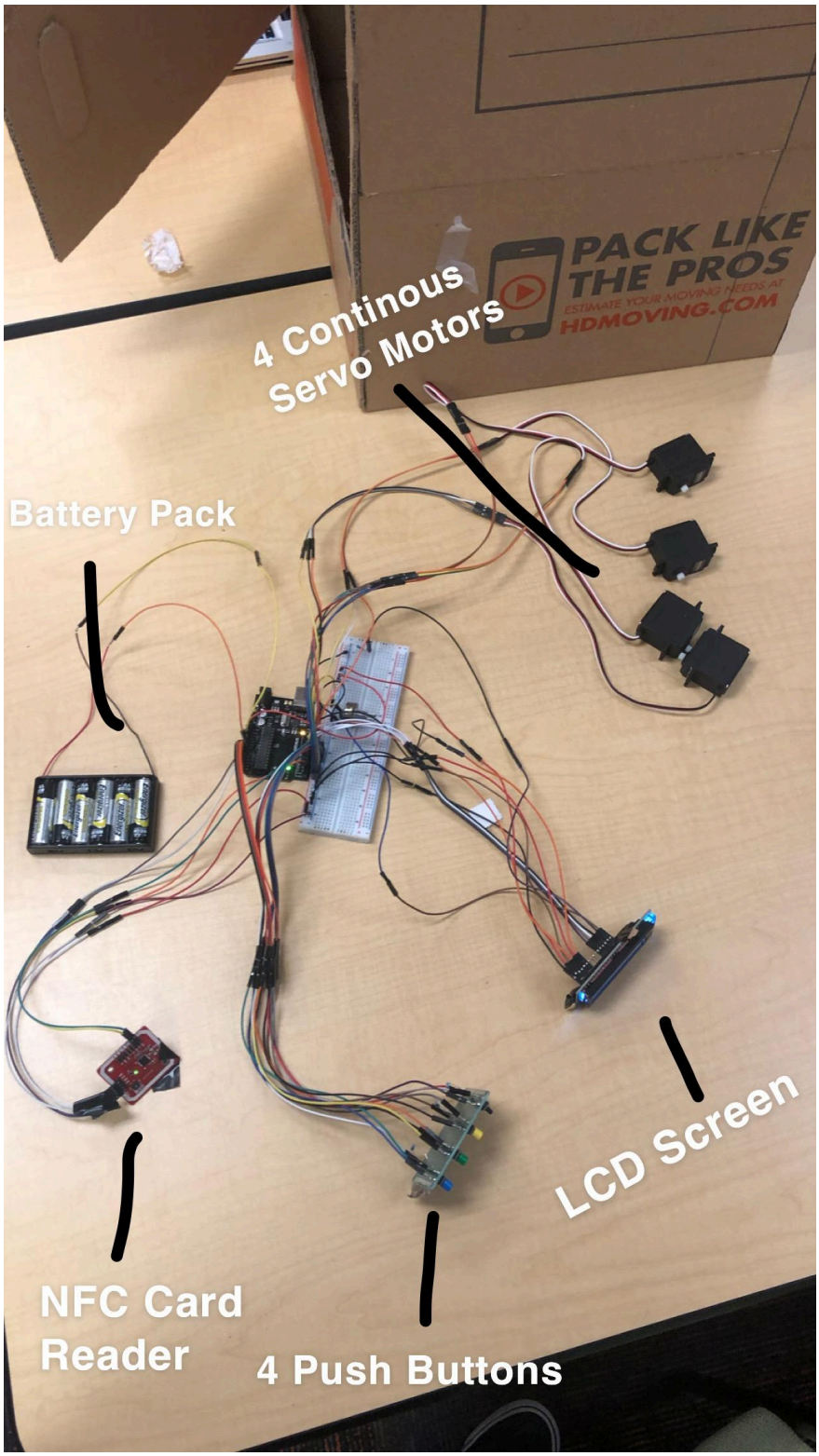
```
    lcd.setCursor(0,1);                                     // Intializes the position of LCD
    lcd.print("$");                                         // Displays "$" on LCD
    lcd.print(Blueprice);                                   // Displays price for Blue
    delay(2000);
    UpdateLCDDisplay();
    lcd.clear();
  }
  if (readA1!=0)                                            // If Green Button pressed
  {
    lcd.clear();                                            // Clears the LCD Screen
    lcd.setCursor(0,0);                                     // Intializes the position of LCD
    lcd.print("The price is");                              // Displays "The price is" on LCD
    lcd.setCursor(0,1);                                     // Intializes the position of LCD
    lcd.print("$");                                         // Displays "$" on LCD
    lcd.print(Greenprice);                                  // Displays price for Green
    delay(2000);
    UpdateLCDDisplay();
    lcd.clear();
  }
  if (readA2!=0)                                            // If Yellow Button pressed
  {
    lcd.clear();                                            // Clears the LCD Screen
    lcd.setCursor(0,0);                                     // Intializes the position of LCD
    lcd.print("The price is");                              // Displays "The price is" on LCD
    lcd.setCursor(0,1);                                     // Intializes the position of LCD
    lcd.print("$");                                         // Displays "$" on LCD
    lcd.print(Yellowprice);                                 // Displays price for Yellow
    delay(2000);
    UpdateLCDDisplay();
    lcd.clear();
  }
  if (readA3!=0)                                            // If Red Button pressed
  {
    lcd.clear();                                            // Clears the LCD Screen
    lcd.setCursor(0,0);                                     // Intializes the position of LCD
    lcd.print("The price is");                              // Displays "The price is" on LCD
    lcd.setCursor(0,1);                                     // Intializes the position of LCD
    lcd.print("$");                                         // Displays "$" on LCD
    lcd.print(Redprice);                                    // Displays price for Red
    delay(2000);
    UpdateLCDDisplay();
    lcd.clear();
  }
  uint8_t success;
  uint8_t uid[] = { 0, 0, 0, 0, 0, 0, 0 };                                  // Buffer to store the returned UID
  uint8_t uidLength;                                                        // Length of the UID (4 or 7 bytes depending on ISO14443A card type)
  success = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A, uid, &uidLength, 100); // Obtains the UID of the card
  if (success) {                                                            // If it reads something, proceed
    nfc.PrintHex(uid, uidLength);
    if (uidLength == 4)                                                     // Identifies the MIFARE Card (A specific type of card)
    {
      uint32_t cardid = uid[0];
      cardid <<= 8;
      cardid |= uid[1];
      cardid <<= 8;
      cardid |= uid[2];                                                     // Assembles the UID into one line.
      cardid <<= 8;
      cardid |= uid[3];
        if (cardid == 4148560155)                                           // Checks if refilling Key was scanned
        {
          cash = cash + 1.75;                                               // Adds a balance of 1.75 to the card
          lcd.clear();                                                      // Clears the LCD Screen
          lcd.setCursor(0,0);                                               // Intializes the position of LCD
          lcd.print("Your new");                                            // Displays "Your new" on LCD
          lcd.setCursor(0,1);                                               // Intializes the position of LCD
          lcd.print("balance is");                                          // Displays "balance is" on LCD
          delay(1000);
          lcd.clear();                                                      // Clears the LCD Screen
          lcd.setCursor(0,0);                                               // Intializes the position of LCD
          lcd.print("$");                                                   // Displays "$" on LCD
          lcd.print(cash);                                                  // Displays the new balance of the card
          delay(1000);
        }
        if (cardid == 2799659557)                                           // Checks if purchasing Card was scanned
        {
          lcd.clear();                                                      // Clears the LCD Screen
          lcd.setCursor(0,0);                                               // Intializes the position of LCD
          lcd.print("Your current");                                        // Displays "Your current" on LCD
          lcd.setCursor(0,1);                                               // Intializes the position of LCD
          lcd.print("balance is");                                          // Displays "balance is" on LCD
          lcd.print(cash);                                                  // Displays balance of card
          delay(3000);
          AlmostPurchase();                                                 // Moves onto the selection of Button
        }
        success = 0;                                                        // If nothing is scanned, it goes back to start
    }
  }
}

void UpdateLCDDisplay()                                                     // Causes the scrolling in the LCD Screen
{
  int iLenOfLargeText = strlen(Text);                                       // Length of Text
  if (iCursor == (iLenOfLargeText - 1) )
  {
    iCursor = 0;
  }
  lcd.setCursor(0,iLineNumber);                                            // Sets where the first letter appears on the LCD

  if (iCursor < iLenOfLargeText - 16)                                       // Moves the text by one letter each time
  {
    for (int iChar = iCursor; iChar < iCursor + 16; iChar++)                // Displays Text on LCD
    {
      lcd.print(Text[iChar]);
    }
  }
  else
  {
    for (int iChar = iCursor; iChar < (iLenOfLargeText - 1); iChar++)
    {
      lcd.print(Text[iChar]);
    }
    for (int iChar = 0; iChar <= 16 - (iLenOfLargeText - iCursor); iChar++)
    {
      lcd.print(Text[iChar]);
    }
  }
  iCursor++;
}

void AlmostPurchase()
{
  lcd.clear();                                                             // Clears the LCD Screen
  lcd.setCursor(0,0);                                                      // Intializes the position of LCD
  lcd.print("Select something.");                                          // Displays "Select something." on LCD
  time_now = millis();
  while (millis() < time_now + 5000)                                       // Creates a timeout function if the person takes too long
  {
  int readA0 = digitalRead(14);                                            // Reads if Blue Button pressed
  int readA1 = digitalRead(15);                                            // Reads if Green Button pressed
  int readA2 = digitalRead(16);                                            // Reads if Yellow Button pressed
  int readA3 = digitalRead(17);                                            // Reads if Red Button pressed
  if (readA0!=0)                                                           // If Blue Button pressed
  {
    if (cash > Blueprice) {                                                // If there's enough balance on the card
      myservo.attach(7);                                                   // Assign Blue's Servo Motor
      myservo.writeMicroseconds(1000);                                     // Turn servo clockwise
```

```
      delay(1000);                                  // Keep servo running clockwise for 1 second
      myservo.writeMicroseconds(1500);              // Stops the servo from moving
      myservo.attach(1);                            // Unassign Blue's Servo Motor
      lcd.clear();                                  // Clears the LCD Screen
      lcd.setCursor(0,0);                           // Intializes the position of LCD
      lcd.print("Purchase Completed!");             // Displays "Purchase Completed!" on LCD
      cash = cash - Blueprice;                      // Subtracts the balance
      delay(2000);
      break;                                        // Goes back to start
    }
    else {
      lcd.clear();                                  // Clears the LCD Screen
      lcd.setCursor(0,0);                           // Intializes the position of LCD
      lcd.print("Not Enough");                      // Displays "Not Enough" on LCD
      lcd.setCursor(0,1);                           // Intializes the position of LCD
      lcd.print("Credit.");                         // Displays "Credit" on LCD
      delay(2000);
      lcd.clear();                                  // Clears the LCD Screen
      break;
    }
  }
  if (readA1!=0)                                    // If Green Button pressed
  {
    if (cash > Greenprice) {                        // If there's enough balance on the card
      myservo.attach(6);                            // Assign Green's Servo Motor
      myservo.writeMicroseconds(1000);              // Turn servo clockwise
      delay(1000);                                  // Keep servo running clockwise for 1 second
      myservo.writeMicroseconds(1500);              // Stops the servo from moving
      myservo.attach(1);                            // Unassign Green's Servo Motor
      lcd.clear();                                  // Clears the LCD Screen
      lcd.setCursor(0,0);                           // Intializes the position of LCD
      lcd.print("Purchase Completed!");             // Displays "Purchase Completed!" on LCD
      cash = cash - Greenprice;                     // Subtracts the balance
      delay(2000);
      break;                                        // Goes back to start
    }
    else {
      lcd.clear();                                  // Clears the LCD Screen
      lcd.setCursor(0,0);                           // Intializes the position of LCD
      lcd.print("Not Enough");                      // Displays "Not Enough" on LCD
      lcd.setCursor(0,1);                           // Intializes the position of LCD
      lcd.print("Credit.");                         // Displays "Credit" on LCD
      delay(2000);
      lcd.clear();                                  // Clears the LCD Screen
      break;
    }
  }
  if (readA2!=0)                                    // If Yellow Button pressed
  {
    if (cash > Yellowprice) {                       // If there's enough balance on the card
      myservo.attach(5);                            // Assign Yellow's Servo Motor
      myservo.writeMicroseconds(1000);              // Turn servo clockwise
      delay(1000);                                  // Keep servo running clockwise for 1 second
      myservo.writeMicroseconds(1500);              // Stops the servo from moving
      myservo.attach(1);                            // Unassign Yellow's Servo Motor
      lcd.clear();                                  // Clears the LCD Screen
      lcd.setCursor(0,0);                           // Intializes the position of LCD
      lcd.print("Purchase Completed!");             // Displays "Purchase Completed!" on LCD
      cash = cash - Yellowprice;                    // Subtracts the balance
      delay(2000);
      break;                                        // Goes back to start
    }
    else {
      lcd.clear();                                  // Clears the LCD Screen
      lcd.setCursor(0,0);                           // Intializes the position of LCD
      lcd.print("Not Enough");                      // Displays "Not Enough" on LCD
      lcd.setCursor(0,1);                           // Intializes the position of LCD
      lcd.print("Not Enough");                      // Displays "Not Enough" on LCD
      lcd.setCursor(0,1);                           // Intializes the position of LCD
      lcd.print("Credit.");                         // Displays "Credit" on LCD
      delay(2000);
      lcd.clear();                                  // Clears the LCD Screen
      break;
    }
  }
  if (readA3!=0)                                    // If Red Button pressed
  {
    if (cash > Redprice) {                          // If there's enough balance on the card
      myservo.attach(4);                            // Assign Red's Servo Motor
      myservo.writeMicroseconds(1000);              // Turn servo clockwise
      delay(1000);                                  // Keep servo running clockwise for 1 second
      myservo.writeMicroseconds(1500);              // Stops the servo from moving
      myservo.attach(1);                            // Unassign Red's Servo Motor
      lcd.clear();                                  // Clears the LCD Screen
      lcd.setCursor(0,0);                           // Intializes the position of LCD
      lcd.print("Purchase Completed!");             // Displays "Purchase Completed!" on LCD
      cash = cash - Redprice;                       // Subtracts the balance
      delay(2000);
      break;                                        // Goes back to start
    }
    else {
      lcd.clear();                                  // Clears the LCD Screen
      lcd.setCursor(0,0);                           // Intializes the position of LCD
      lcd.print("Not Enough");                      // Displays "Not Enough" on LCD
      lcd.setCursor(0,1);                           // Intializes the position of LCD
      lcd.print("Credit.");                         // Displays "Credit" on LCD
      delay(2000);
      lcd.clear();                                  // Clears the LCD Screen
      break;
    }
    delay(2000);
  }
}
}
```

5. Circuit Schematics



6. Circuit Prototype



7. Testing

When we were figuring out how each component worked we decided to first test out how to command a servo motor based on simply pressing a button. During this test, we were having trouble getting the motor to spin but were eventually able to. From there, our second test came in when we wired the NFC card and LCD screen. We had to test showing text then after tapping the card, pressing a button which would subsequently spin the motor. Our process was simply adding a new component each time to the process. From there we added all the buttons, servos, and additional if and then statements. We also had to test the range that we had to spin the motors in order to dispense a single snack rather than many. Our final test came in when we simulated an actual vending machine by displaying prices then trying to buy something without actually having money; then actually having money allowed us to get a snack.