Electromagnetic
Bodybuilder

Final Report | ECE 445

Harrison Kim | Michael Liu
Group 50

TA: Jackson Lenz
May 4, 2016
Abstract

The electromagnetic bodybuilder is a system that utilizes electromagnetic force to build muscle instead of traditional weights that are dependent on mass and gravity. Our system replicates a bicep curl motion but can be expanded to larger ones such as a bench press. The system is much lighter and more easily transportable than sets of weights. It can be used wherever there is a wall outlet in America and space. The system also gives the user the freedom to choose precise weights they would like to simulate. Lastly, the system does not depend on gravity so it can be used in space where there is no gravity.
Contents

1. Introduction .................................................................................................................. 1
   1.1. Statement of Purpose .......................................................................................... 1
   1.2. Objectives ........................................................................................................... 1
   1.3. Block Diagram and Descriptions ....................................................................... 1
       1.3.1. Power Supply .............................................................................................. 2
       1.3.2. Solenoid ...................................................................................................... 2
       1.3.3. User Display ............................................................................................... 2
2. Design .......................................................................................................................... 4
   2.1. Design Procedure .................................................................................................. 4
       2.1.1. Power Supply .............................................................................................. 4
       2.1.2. Solenoid ...................................................................................................... 4
       2.1.3. User Display ............................................................................................... 5
   2.2. Design Details ....................................................................................................... 5
       2.2.1. Power Supply .............................................................................................. 5
       2.2.2. Solenoid ...................................................................................................... 6
       2.2.3. User Display ............................................................................................... 8
3. Verification .................................................................................................................... 10
   3.1. Power Supply ...................................................................................................... 10
   3.2. Solenoid ............................................................................................................ 11
   3.3. User Display ...................................................................................................... 11
4. Costs ............................................................................................................................ 12
   4.1. Parts List ............................................................................................................. 12
   4.2. Labor .................................................................................................................. 12
5. Conclusion .................................................................................................................... 13
   5.1. Accomplishments .............................................................................................. 13
   5.2. Uncertainties ....................................................................................................... 13
   5.3. Ethical Considerations ....................................................................................... 13
   5.4. Safety Considerations ....................................................................................... 13

References ....................................................................................................................... 15

Appendix A Requirements and Verification Table ......................................................... 16
Appendix B Arduino Weight Meter IDE ..................................................................... 18
1. Introduction

1.1. Statement of Purpose
There are several components in a free weight system that could be improved. First, having to constantly replace weights on the bar and organizing them are tedious actions in the bench press process. As well as being a tedious action, buying sets of weights that can only increment by 5 lbs at a time can be very costly. We believe that our electromagnetic bodybuilding system will be able to overcome these inconveniences and that this system could potentially be the future of body building systems.

1.2. Objectives
The goal of this project is to replicate a free weight system using electromagnetic force with easy to use controls for the user. Some benefits of an electromagnetic bodybuilding system include:

- Eliminates the tedious process of switching weights
- The amount of force will be able to be changed more precisely than current free weights
- Having to buy sets of weights is no longer necessary
- Will maintain same amount of force in places with lower or no gravity

1.3. Block Diagram and Descriptions
Figure 1 below shows an overview of the entire electromagnetic bodybuilding mechanism. In the following sections, each block’s function will be discussed.

---

![Block Diagram of Entire System](image.png)

Figure 1: Block diagram of entire system
1.3.1. **Power Supply**

The purpose of the power supply is to convert an input 120Vac wall voltage to a variable DC voltage output. The power supply is composed of several components including:

- **5:1 Transformer:** This device reduces the input ac voltage to an output voltage that is one-fifth that of the input voltage.
- **Rectifier:** Allows current in only one direction so that only the positive current portion of the AC voltage is passed to the next part of the power supply.
- **Voltage Regulator (LM317T):** Smooth out the AC voltage with the capacitor and allows variable DC output voltage depending on the resistor at the output and adjust pin. The output voltage determines the emitter voltage of the power transistors so that current is not strained on the LM317 itself.
- **Potentiometer:** The potentiometer at the adjust pin is used as a user control device to control the output voltage of the LM317 voltage regulator. An increase in resistance from the potentiometer constitutes an increase in output voltage while a decrease in resistance constitutes a decrease in output voltage from the LM317.
- **Power Transistors (2N3055):** The transistors allow current from the input that is controllable depending on the output voltage of the LM317T to the transistor bases. The collectors of the transistors are connected to the output of the rectifier and the emitters are the output voltage of the system. The transistors act as heat sinks and more can be added in parallel for more heat distribution.

1.3.2. **Solenoid**

The solenoid is responsible for the strength and direction of the electromagnetic field the user has to overcome. The power supply’s output voltage determines the current through the solenoid. The solenoid contains an iron core which attracts an iron rod in which the user will separate from each other to experience force.

1.3.3. **User Display**

In order for the user to see what weight they are simulating, an Arduino is used to read the resistance across the potentiometer and to display the expected weight based on the read resistance. The user display consists of two main components:

- **Weight Meter:** The Arduino reads the resistance by a voltage divider equation when the potentiometer is not connected in the circuit. In order to calculate the expected weight from the read resistance, multiple test trials were taken at different resistances to determine the expected weight.
at those resistance, then a fourth degree polynomial equation was taken for the line of best fit and inserted into the code for calculation.

- **Seven Segment Display**: Two seven segment displays are used to display the expected weight calculated from the weight meter. 14 pins are used from the Arduino and a count is used for the tens and ones digits from the calculated weight in order to determine which LED’s are lit for each digit.
2. Design

2.1. Design Procedure

2.1.1. Power Supply

When designing the power supply, the largest concern was the circuit being able to transfer and output large currents of up to 3A. The first circuit consisted of using multiple LM317T voltage regulators in parallel to distribute current, however current was not even distributed and certain LM317T regulators would experience more current and heat that others. Not only this, but the potentiometer experienced temperatures above 200°C so the only way to solve this was to hand wind a nichrome potentiometer to give us the exact resistances needed. Although this circuit worked, it was not very cost or space efficient due to the LM317T’s and electrolytic capacitors for each. The nichrome wire for the potentiometer was not cost efficient as well. Lastly, because the potentiometer would be experiencing high current, the Arduino would not be able to read the resistance across it due to heat transfer through the wire.

The alternative design, which was implemented for the final product, of the power supply utilized the LM317T regulator as a means to control output voltage instead of supplying it to the solenoid. This circuit uses a single LM317T and a single capacitor. Also all of the LM317T terminals experience current of less than 100mA each. The output voltage of the LM317T constitutes the emitter voltage of the transistors in parallel so the transistors would be strained with the high current instead of the LM317T.

2.1.2. Solenoid

When designing the solenoid, we had to determine which factors of the magnetic force we should control and which ones should be left constant. In Equation (1), it is seen that the magnetic force is dependent on the number of coil turns (N), the current through the solenoid (I), the magnetic permeability of the core (µ), the cross sectional area of the solenoid (A), and the length of the solenoid (l). An infinite number of designs for the solenoid can be implemented through proper selection of these parameters, but our design choices are outlined in section 2.2.2.

\[
F = \frac{N^2 I^2 \mu A}{2l^2}
\]  
(1)
2.1.3. **User Display**
The user display plays an integral part of the operation for the user, so when designing the display, we needed a simple, intuitive system for the user to see what weight they will be experiencing. For simplicity, we decided to use an Arduino Uno and two 7-segment LEDs. The Arduino IDE measures the resistance of the potentiometer, converts it to the weight that the user will experience and display that weight onto the 7-segment LEDs.

2.2. **Design Details**

2.2.1. **Power Supply**
The power supply converts 120Vac to variable DC voltage and the circuit schematic is shown in Figure 2. The open circuit voltage ranges from 24Vdc to 37Vdc.

![Power supply circuit schematic](image)

**Figure 2:** Power supply circuit schematic

The transformer is 5:1 and converts 120Vac to 24Vac. The circuit represents this transformer as an inductance ratio of 25:1 which can be found from Equation (2).

\[
\frac{N_1}{N_2} = \frac{L_1}{L_2}
\]  

Equation (3) below determines the output voltage of the LM317 and the whole circuit. Although it is possible to yield infinitely large voltages depending on \( R_{\text{adj}} \) and \( R_{\text{out}} \), the max output voltage of the LM317 is around 37V, hence our max output voltage of 38V.

\[
V_{\text{out}} = 1.25(1 + \frac{R_{\text{adj}}}{R_{\text{out}}}) + I_{\text{adj}}R_{\text{adj}}
\]  

(3)
This output voltage of the LM317 constitutes the emitter voltage of the 2N3055 power transistors. The transistors are rated up to 115W, 60V, and 15A while also having an operating temperature range of -65° to 200°C. Although a single transistor would be enough, the heat dissipated at this transistor would interfere with other components and would melt insulation on wire if they touched it so multiple transistors are placed in parallel to distribute heat. Figure 3 below shows the PCB schematic in Eagle CAD of the power supply circuit.

![Eagle CAD PCB schematic of power supply circuit](image)

**Figure 3:** Eagle CAD PCB schematic of power supply circuit

### 2.2.2. Solenoid

When choosing the parameter values of the solenoid, it is clear that the most feasible parameter to control is the current, therefore we chose to set all the remaining parameters at fixed values. We then took into consideration the maximum force required of our system and the magnetic permeability of the available soft iron core, while setting the remainder of the values according to the materials available to us. After taking these requirements into consideration, we concluded with \( N = 1500 \) turns, \( \mu_r = 2928 \) (the relative permeability of the iron core), core radius \( r = 0.5 \) inches, and \( l = 4 \) inches.

When choosing the material of the core, we had to ensure that it would be able to quickly magnetize with the iron core in the solenoid. We based our decision to choose a soft iron core due to the thin magnetic hysteresis curve of soft ferromagnetic materials as seen from Figure 4. Since the material can saturate easily, it is ideal to use for the solenoid.
With the parameters chosen for the solenoid, we plotted the relationship, dictated by Equation 1 from section 2.1.2, between force (lbs) and the current (A) in Figure 4. It is important to observe that the relationship between force and current is nonlinear. This nonlinear relationship reveals that at higher weights (i.e. 20-35 lbs) it takes smaller increments of current to increase force by one pound than it would at lower weights (i.e. 1-19 lbs) since the factor of current is squared.

The output voltage of the power supply is decreased due to voltage drop from the solenoid load being added. The range of operating voltages with the added load
was expected to be from 0V to 12.5Vdc and is shown in Figure 5. Since the resistance of the solenoid is 5.6Ohms, achieving the goal of 2A current through the solenoid is possible as seen by a simple calculation of Ohm’s Law in Equation (4).

\[
I = \frac{V}{R} = \frac{12.5}{5.6} = 2.2083A > 2A
\]  

(4)

Figure 6: Current from power supply to solenoid

2.2.3. User Display

The user display utilizes an Arduino Uno to measure the resistance of at the potentiometer, converts this resistance to the weight the user will experience during the exercise, and finally outputs this weight to display on two 7-segment LEDs. The setup of this circuit is seen in Figure 6. We implement the ohmmeter portion of the Arduino as described in Reference [1]. The Arduino IDE measures the resistance using a voltage divider, comparing the resistance of the potentiometer at any given moment with the resistance of a set resistor. In our case, the fixed resistor value is 330 Ω. This resistor value was chosen because due to the nature of the code, the closer the fixed resistance is to the variable resistance, the more accurate the reading. Our potentiometer resistance ranges from 0-500 Ω, so we chose a resistance within this range to keep the measurements as accurate as possible.
Once the resistance of the potentiometer has been measured, the IDE uses an equation whose input is the resistance of the potentiometer and whose output is the weight in pounds the user will experience. This equation was found by taking several test trials at different resistance levels to measure the real weight simulated. This data was then plotted in excel to find a best fit curve for the data; a fourth degree polynomial trend was the closest fit to the data and the equation calculated in excel for this trend is seen in Equation (5). Finally, the IDE outputs the weight that the user will experience to the two 7-segment LEDs for the user to see. All of the code written for the Arduino to display the weight to the user is found in Appendix B.

\[
\text{Weight} = 3 \times 10^{-6} R^3 - 1 \times 10^{-4} R^2 + 0.125R - 0.8173
\]  

(5)
3. Verification

3.1. Power Supply

In order to verify the operation of the power supply, we simply tested the output voltage of the power supply when no load was added. An oscilloscope was used to probe Vout as we adjusted the resistance across the potentiometer. It can be seen that the minimum output voltage is 24.1Vdc (Figure 7) and the maximum output voltage is 38.1Vdc (Figure 8). Also, the output voltage can be any value between those voltages, depending on the potentiometer resistance. In addition to verifying variable voltage, the voltage is clearly DC, with no spikes in the waveform.

Figure 8: Minimum output voltage of power supply

Figure 9: Maximum output voltage of power supply
3.2. Solenoid

The operation of the solenoid was verified by its ability to increase and decrease force depending on the current. We initially used the bench DC power supply across the solenoid and verified that the experienced force did increase with increasing voltage. We then used our power supply to determine this requirement since we verified it had already been working and testing with the power supply would also allow us to take data for the conversion equation used to calculate weight from potentiometer resistance. Using a luggage weight, we recorded the weights at different resistances across the potentiometer and plotted the relationship which is shown in Figure 9.

![Figure 9: Pot Resistance vs. Weight](image)

From the recorded points, it can be seen that weight does in fact increase with increasing potentiometer resistance and increasing output voltage. We also used the multimeter at max output voltage and observed an output voltage of 13V. Since the resistance of the solenoid is 5.6Ohms, the current at this voltage was 2.3A which meets the requirement of being below 3A.

3.3. User Display

The weight meter was simply verified by observing the weight displayed on the luggage weight and the weight displayed on the 7-segment LED’s. Our requirement was that the weight displayed on the weight meter was within 1lb of error displayed on the luggage weight.
4. **Costs**

4.1 **Parts List**

<table>
<thead>
<tr>
<th>Part</th>
<th>Quantity</th>
<th>Price(unit)</th>
<th>Price(all)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2N3055 Transistor</td>
<td>3</td>
<td>$3.00</td>
<td>$9.00</td>
</tr>
<tr>
<td>470µF Capacitor</td>
<td>1</td>
<td>$0.39</td>
<td>$0.39</td>
</tr>
<tr>
<td>5:1 Transformer</td>
<td>1</td>
<td>$14.00</td>
<td>$14.00</td>
</tr>
<tr>
<td>7-Segment LED</td>
<td>2</td>
<td>$0.95</td>
<td>$1.90</td>
</tr>
<tr>
<td>Arduino Uno</td>
<td>1</td>
<td>$11.53</td>
<td>$11.53</td>
</tr>
<tr>
<td>Diode</td>
<td>5</td>
<td>$0.16</td>
<td>$0.80</td>
</tr>
<tr>
<td>LM317T Voltage Regulator</td>
<td>1</td>
<td>$0.52</td>
<td>$0.52</td>
</tr>
<tr>
<td>Magnet Copper Wire Spool 22AWG</td>
<td>3</td>
<td>$12.00</td>
<td>$36.00</td>
</tr>
<tr>
<td>Paracord</td>
<td>1</td>
<td>$3.00</td>
<td>$3.00</td>
</tr>
<tr>
<td>PCB (Printed Circuit Board)</td>
<td>1</td>
<td>$33.00</td>
<td>$33.00</td>
</tr>
<tr>
<td>PCB Box</td>
<td>1</td>
<td>$12.70</td>
<td>$12.70</td>
</tr>
<tr>
<td>Plywood Base</td>
<td>1</td>
<td>$15.00</td>
<td>$15.00</td>
</tr>
<tr>
<td>Potentiometer</td>
<td>1</td>
<td>$4.27</td>
<td>$4.27</td>
</tr>
<tr>
<td>PVC Pipe</td>
<td>1</td>
<td>$2.35</td>
<td>$2.35</td>
</tr>
<tr>
<td>Resistors</td>
<td>5</td>
<td>$0.15</td>
<td>$0.75</td>
</tr>
<tr>
<td>Spring</td>
<td>2</td>
<td>$2.50</td>
<td>$5.00</td>
</tr>
<tr>
<td>Soft Iron Rod</td>
<td>2</td>
<td>$10.00</td>
<td>$20.00</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td><strong>$170.21</strong></td>
</tr>
</tbody>
</table>

4.2 **Labor**

<table>
<thead>
<tr>
<th>Name</th>
<th>Hourly Rate</th>
<th>Hours Invested</th>
<th>Total = Rate * Hours * 2.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harrison Kim</td>
<td>$40</td>
<td>200</td>
<td>$20,000</td>
</tr>
<tr>
<td>Michael Liu</td>
<td>$40</td>
<td>200</td>
<td>$20,000</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td><strong>$40,000</strong></td>
</tr>
</tbody>
</table>

**Grand Total = $170.21 + $40,000 = $40,170.21**
5. Conclusions

5.1 Accomplishments
We were able to complete the complete system as designed. We were able to simulate weights between 2 and 38 pounds using electromagnetic force for a bicep curl motion. We successfully made a power supply that uses wall AC voltage and converts it to variable DC. We also designed a solenoid that can simulate the desire force of 0 to 40 pounds without having to supply current above 3A. Lastly, we completed a weight meter that displays the expected weight that is to be simulated on two seven-segment LEDs.

5.2 Uncertainties
Although we completed all of the requirements of our design, there are still some uncertainties in the product. Due to time constraints, we were unable to test what might happen to the system if left on for long periods of time. Also, because the weight meter was added later in our design, we were unable to supply power for it with the power supply circuit itself, therefore the Arduino’s battery would have to be recharged separately.

5.3 Ethical Considerations
We abide to the IEEE Code of Ethics from reference [4] and address the points of concern.

“to avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist;”

We have stated which point are of most concern in our system, particularly heating from long periods of use.

“to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others;”

Because we had time and cost constraints, our design certainly has the potential to be more efficient and convenient for users. We are open to all suggestions and criticisms.

“to avoid injuring others, their property, reputation, or employment by false or malicious action;”

We made sure that none of the wires in our circuits are exposed and that all of the circuitry is contained in a box so that user does not face direct harm from the system.

5.4 Safety Considerations
The system uses high currents that can be dangerous to users if they come in direct contact with them. All of the wires in our circuit are insulated however if they are to be
exposed, they can be very dangerous of users. Also, the springs are vital to giving a softer motion of the bicep curl. If the springs are removed, the user can put strain on their joints and muscles so these should never be removed.
References


[8] “Solenoid,” Georgia State University. [Online]. Available at: http://hyperphysics.phy-astr.gsu.edu/hbase/magnetic/solenoid.html
## Appendix A

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Verification</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AC-DC Converter</strong>&lt;br&gt;1. $V_{out} = 37\text{VDC} \pm 1.5\text{V}$ with 500 Ohms at lower resistor and $V_{out} = 24\text{VDC} \pm 1.5\text{V}$ with 50 Ohms at lower resistor, when used with transformer and wall 120VAC, 60Hz.&lt;br&gt;2. Output voltage increases and decreases with potentiometer resistance changes.</td>
<td><strong>AC-DC Converter</strong>&lt;br&gt;1. Verification process for item 1:&lt;br&gt; a. Connect input terminal of rectifier circuit to waveform generator outputting at 20 Vpp, 60 Hz&lt;br&gt; b. Connect output of the converter circuit to oscilloscope.&lt;br&gt; c. Set resistance at the adjust pin of LM317 to 5 kOhm and verify that the output of the converter circuit is 1.2V and that the output is DC signal.&lt;br&gt;2. Verification process of item 2:&lt;br&gt; a. Connect input of converter circuit to waveform generator outputting at 24 VRMS, 60 Hz&lt;br&gt; b. Starting from 0 Ohms on potentiometer at the LM317 adjust pin, increase resistance by 0.5 Ohm increments.&lt;br&gt; c. Verify that output voltage of the converter can supply a voltage between 0 V and 8 V&lt;br&gt;3. Verification process of item 3:&lt;br&gt; a. Connect function generator at 20 Vpp to input of circuit.&lt;br&gt; b. Connect output of circuit to multimeter, measure DC voltage.&lt;br&gt; c. Verify that output voltage increases when the resistance across the potentiometer increases and voltage decreases when the potentiometer resistance decreases.</td>
<td>25</td>
</tr>
<tr>
<td>Arduino Weight Meter</td>
<td>Arduino Weight Meter</td>
<td>15</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------------------</td>
<td>----</td>
</tr>
<tr>
<td>1. Arduino displays weight resistance relationship as displayed on trial table ± 1lb</td>
<td>1. Verification process for item 1:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>a. Connect the potentiometer used at the LM317 adjust node to input of weight meter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>b. Start potentiometer at lowest resistance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>c. Verify the weight read on the meter reflects the trial table of weight/resistance relationship</td>
<td></td>
</tr>
<tr>
<td></td>
<td>d. Increase potentiometer gradually by 50 Ohm at a time and verify that the weight on the meter matches the trial table.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Solenoid</th>
<th>Solenoid</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Wire functions for $I_{\text{max}} \leq 3\text{A}$</td>
<td>1. Verification process for item 1:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>a. Connect positive terminal of dc power supply to solenoid terminals.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>b. Starting from 0 V, increase voltage supply in increments of 0.5 V until 6 V is reached</td>
<td></td>
</tr>
<tr>
<td></td>
<td>c. As voltage is increased, verify that current never exceeds 1.5 A (by Ohm’s law, this should translate to staying under 3 A for 12 V).</td>
<td></td>
</tr>
</tbody>
</table>
Appendix B
(read from left column to right column)

#define A 7 //Left 14
#define B 8 //13
#define C 2 //8
#define D 3 //7
#define E 4 //6
#define Ff 5 //1
#define G 6 //2
#define AA 9 //Right 14
#define B2 10 //13
#define C2 11 //8
#define D2 12 //7
#define E2 13 //6
#define F2 15 //1
#define G2 16 //2

int analogPin= 0;
int raw= 0;
int Vin= 5;
float Vout= 0;
float R1= 326;
float R2= 0;
float buffer= 0;
int weight = 0;
int tempweight = 0;
int count = 0;

void setup()
{
    //Setup our pins
    pinMode(A, OUTPUT);
    pinMode(B, OUTPUT);
    pinMode(C, OUTPUT);
    pinMode(D, OUTPUT);
    pinMode(E, OUTPUT);
    pinMode(Ff, OUTPUT);
    pinMode(G, OUTPUT);
    pinMode(AA, OUTPUT);
    pinMode(B2, OUTPUT);
    pinMode(C2, OUTPUT);
    pinMode(D2, OUTPUT);
    pinMode(E2, OUTPUT);
    pinMode(F2, OUTPUT);
    pinMode(G2, OUTPUT);
    Serial.begin(9600);  //Begin serial communication
}

void loop()
{
    raw= analogRead(analogPin);
    if(raw)
    {
        buffer= raw * Vin;
        Vout= (buffer)*1024.0;
        buffer= (Vin/Vout) -1;
        R2= R1 * buffer;
        weight = 3e-8*pow(R2,3) -
           0.0001*pow(R2,2) + 0.1252*R2 -
           0.8173; //Weight Equation
        Serial.print("Vout: ");
        Serial.println(Vout);
        Serial.print("R2: ");
        Serial.println(R2);
        Serial.print("Weight: ");
        Serial.println(weight);
        tempweight = weight;
        while(tempweight >= 10)
        {
            tempweight = tempweight - 10;
            count++;
        }
    }
    switch(tempweight){
    case 0:
        zero();
        break;
    case 1:
        one();
        break;
    case 2:
        two();
        break;
    case 3:
        three();
        break;
    case 4:
        four();
        break;
    case 5:
        five();
        break;
    case 6:
        six();
        break;
    case 7:
        seven();
        break;
    case 8:
        eight();
        break;
    case 9:
        nine();
        break;
    }
    switch(count){
    case 0:
        delay(200);
    }
    switch(tempweight){
    case 0:
        zero2();
        break;
    case 1:
        one2();
        break;
    case 2:
        two2();
        break;
    case 3:
        three2();
        break;
    case 4:
        four2();
        break;
    case 5:
        five2();
        break;
    case 6:
        six2();
        break;
    case 7:
        seven2();
        break;
    case 8:
        eight2();
        break;
    case 9:
        nine2();
        break;
    }
    switch(count){
    case 0:
        zero();
        break;
    case 1:
        one();
        break;
    case 2:
        two();
        break;
    case 3:
        three();
        break;
    case 4:
        four();
        break;
    case 5:
        five();
        break;
    }
    switch(tempweight){
    case 0:
        zero();
        break;
    case 1:
        one();
        break;
    case 2:
        two();
        break;
    case 3:
        three();
        break;
    case 4:
        four();
        break;
    case 5:
        five();
        break;
    case 6:
        six();
        break;
    case 7:
        seven();
        break;
    case 8:
        eight();
        break;
    case 9:
        nine();
        break;
    }
}

void one()
{
    //Displays 1
    digitalWrite(D, LOW);
    digitalWrite(E, LOW);
    digitalWrite(Ff, LOW);
    digitalWrite(G, LOW);
    digitalWrite(A, LOW);
}

void two()
{
    //Displays 2
    digitalWrite(D, HIGH);
    digitalWrite(E, HIGH);
    digitalWrite(Ff, LOW);
    digitalWrite(G, HIGH);
    digitalWrite(A, LOW);
}

void three()
{
    //Displays 3
    digitalWrite(D, HIGH);
    digitalWrite(E, HIGH);
    digitalWrite(Ff, LOW);
    digitalWrite(G, HIGH);
    digitalWrite(A, HIGH);
}

void four()
{
    //Displays 4
    digitalWrite(D, LOW);
    digitalWrite(E, LOW);
    digitalWrite(Ff, HIGH);
    digitalWrite(G, HIGH);
    digitalWrite(A, LOW);
}
void five()
{
    digitalWrite(B, HIGH);
    digitalWrite(C, HIGH);
}

void six()
{
    digitalWrite(B, HIGH);
    digitalWrite(C, HIGH);
}

void seven()
{
    digitalWrite(D, LOW);
    digitalWrite(E, LOW);
    digitalWrite(F, LOW);
    digitalWrite(G, LOW);
    digitalWrite(A, HIGH);
    digitalWrite(B, HIGH);
    digitalWrite(C, HIGH);
}

void eight()
{
    digitalWrite(D, HIGH);
    digitalWrite(E, HIGH);
    digitalWrite(F, HIGH);
    digitalWrite(G, HIGH);
    digitalWrite(A, HIGH);
    digitalWrite(B, HIGH);
    digitalWrite(C, HIGH);
}

void nine()
{
    digitalWrite(D, HIGH);
    digitalWrite(E, LOW);
    digitalWrite(F, HIGH);
    digitalWrite(G, HIGH);
    digitalWrite(A, HIGH);
    digitalWrite(B, HIGH);
    digitalWrite(C, HIGH);
}

void zero()
{
    digitalWrite(B, HIGH);
    digitalWrite(C, HIGH);
    digitalWrite(D, LOW);
    digitalWrite(E, LOW);
    digitalWrite(F, LOW);
    digitalWrite(G, LOW);
    digitalWrite(A, HIGH);
    digitalWrite(B, HIGH);
    digitalWrite(C, HIGH);
}

void one2()
{
    digitalWrite(D2, LOW);
    digitalWrite(E2, LOW);
    digitalWrite(F2, LOW);
    digitalWrite(G2, LOW);
    digitalWrite(AA, LOW);
    digitalWrite(B2, HIGH);
    digitalWrite(C2, HIGH);
}

void two2()
{
    digitalWrite(D2, HIGH);
    digitalWrite(E2, HIGH);
    digitalWrite(F2, LOW);
    digitalWrite(G2, HIGH);
    digitalWrite(AA, HIGH);
    digitalWrite(B2, HIGH);
    digitalWrite(C2, LOW);
}

void three2()
{
    digitalWrite(D2, HIGH);
    digitalWrite(E2, LOW);
    digitalWrite(F2, LOW);
    digitalWrite(G2, HIGH);
    digitalWrite(AA, HIGH);
    digitalWrite(B2, HIGH);
    digitalWrite(C2, HIGH);
}

void four2()
{
    digitalWrite(D2, LOW);
    digitalWrite(E2, LOW);
    digitalWrite(F2, HIGH);
    digitalWrite(G2, HIGH);
    digitalWrite(AA, LOW);
    digitalWrite(B2, HIGH);
    digitalWrite(C2, HIGH);
}

void five2()
{
    digitalWrite(D2, HIGH);
    digitalWrite(E2, LOW);
    digitalWrite(F2, HIGH);
    digitalWrite(G2, HIGH);
    digitalWrite(AA, HIGH);
    digitalWrite(B2, LOW);
    digitalWrite(C2, HIGH);
}

void six2()
{
    digitalWrite(D2, HIGH);
    digitalWrite(E2, HIGH);
    digitalWrite(F2, HIGH);
    digitalWrite(G2, HIGH);
    digitalWrite(AA, HIGH);
    digitalWrite(B2, LOW);
    digitalWrite(C2, HIGH);
}

void seven2()
{
    digitalWrite(D2, LOW);
    digitalWrite(E2, LOW);
    digitalWrite(F2, LOW);
    digitalWrite(G2, HIGH);
    digitalWrite(AA, HIGH);
    digitalWrite(B2, HIGH);
    digitalWrite(C2, HIGH);
}

void eight2()
{
    digitalWrite(D2, HIGH);
    digitalWrite(E2, HIGH);
    digitalWrite(F2, HIGH);
    digitalWrite(G2, HIGH);
    digitalWrite(AA, HIGH);
    digitalWrite(B2, HIGH);
    digitalWrite(C2, HIGH);
}

void nine2()
{
    digitalWrite(D2, HIGH);
    digitalWrite(E2, LOW);
    digitalWrite(F2, HIGH);
    digitalWrite(G2, HIGH);
    digitalWrite(AA, HIGH);
    digitalWrite(B2, HIGH);
    digitalWrite(C2, HIGH);
}

void zero2()
{
    digitalWrite(D2, HIGH);
    digitalWrite(E2, LOW);
    digitalWrite(F2, HIGH);
    digitalWrite(G2, HIGH);
    digitalWrite(AA, LOW);
    digitalWrite(B2, HIGH);
    digitalWrite(C2, HIGH);
}