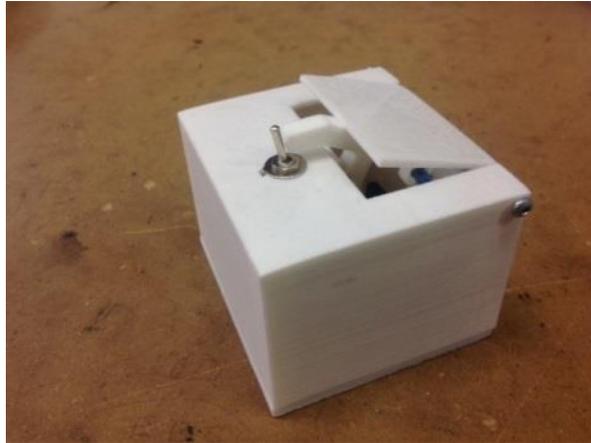


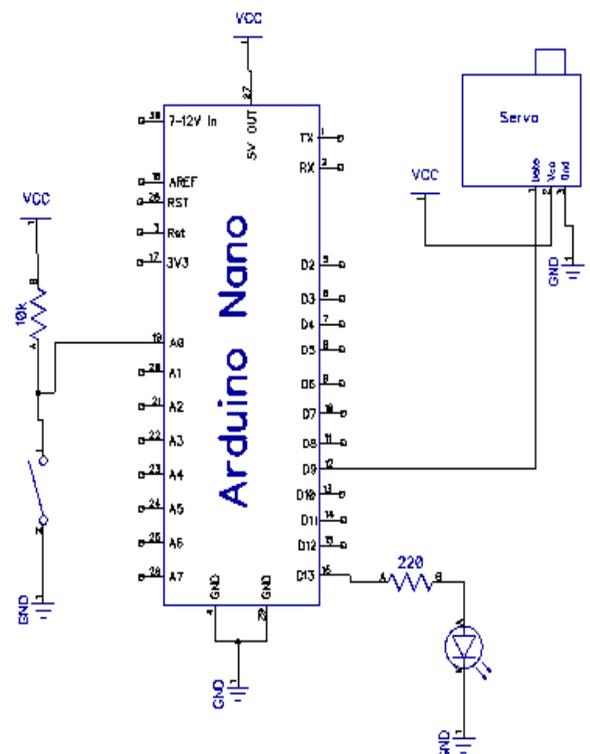
The “USELESS BOX”



The useless box is exactly what it implies.... A project that is pretty much “Useless” and is made for pure entertainment. You are going to go through the process of building this project from breadboarding and programming it using Arduino, to making a circuit board and finally designing the case in 3D and printing it out on a 3D printer. In the end, you will have a “Useless Box”

Procedure:

- 1) Breadboard the schematic to the right. You will need to get a servo motor from the teacher.
- 2) Once you have breadboarded everything, start the BASIC coding section to make sure the 3 components all work (LED, Switch and Servo). Here are the suggestions to get you started using the Arduino software examples;
 - Open up the “Blink” example under 01.Basics. Does the LED blink on Pin 13? If so, go to the next step.
 - Open up the “DigitalReadSerial” example under 01.Basics, change the line; `int pushButton = 2;` to `int pushButton = A0;` to match the pins you are using in the schematic. Upload the code. Open up the serial monitor and see if you get a 1 or 0 as you change the state of the switch (connect, disconnect wires). If so, go to the next step.
 - Open up the “Sweep” example under Servo. Upload the code. The servo should go back and forth from 0 degrees to 180 degrees.



DO NOT CONTINUE UNLESS ALL 3 COMPONENTS ABOVE ARE WORKING WITH CODE!!

3) Now that you have the 3 individual parts working with code you are going to combine them. Start a “New” blank Sketch.

- Copy EVERYTHING from all 3 programs (Blink, DigitalReadSerial and Sweep) that is above the void setup() OR in void setup(). It should look like the example to the right once all 3 parts of the programs are combined.
- Copy in the structure of the DigitalRead Serial” as you are going to read the “state” of the button. See below.

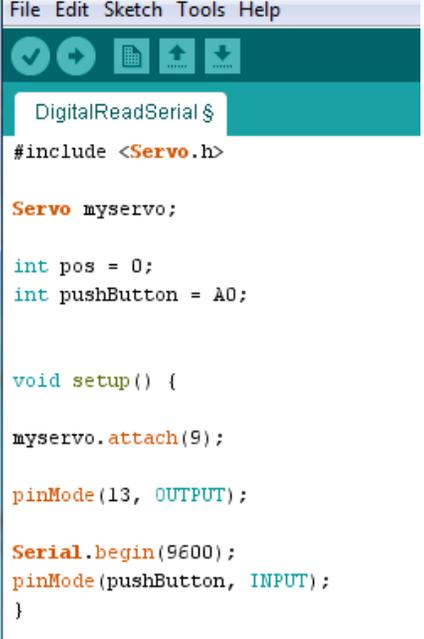
```
void loop() {
  int buttonState = digitalRead(pushButton);
  Serial.println(buttonState);
}
```

- Next, using what you learned in Lab 3 of Arduino Breadboarding, bring in 2 “IF” statements to react to the “state” of the pushbutton. See below.

```
void loop()
{
  int buttonState = digitalRead(pushButton);
  Serial.println(buttonState);

  if (buttonstate == 1)
  {
    digitalWrite(13, HIGH);
  }
  if (buttonstate == 0)
  {
    digitalWrite(13, LOW);
  }
}
```

- Upload the code now and test. Does the LED turn ON and OFF as you change the “state” of the switch? If so, continue on.
- Now, bring in the Servo Code from the example “Sweep”. If you look at the code, you will see that there is what is called a “For Loop” that is being used. This code is simply starting at 0 and advancing 1 “pos” at a time from 0 to 180 with a 15ms delay between each advancement. The bit of code you are interested in is the *myservo.write (pos);* If you replace “pos” with a number between 0 and 180, the servo will go to that position.
- Copy the line *myservo.write (pos);* into both “IF” statements in your program. Change the “pos” in each “IF” statement to 0 and 180, respectively.



```
File Edit Sketch Tools Help
DigitalReadSerial $
#include <Servo.h>

Servo myservo;

int pos = 0;
int pushButton = A0;

void setup() {
  myservo.attach(9);
  pinMode(13, OUTPUT);

  Serial.begin(9600);
  pinMode(pushButton, INPUT);
}
```

```
for(pos = 0; pos <= 180; pos += 1)
{
  myservo.write(pos);
  delay(15);
}
```

```
if (buttonstate == 1)
{
  digitalWrite(13, HIGH);
  myservo.write(0);
}
```

- Upload the code. The Servo should move when you change the “state” of the button and the LED should turn ON and OFF as well still. If so, programming is complete.

```

#include <Servo.h>
#define ServoPin 9
#define ButtonPin 2
#define LEDPin 10

Servo myServo;
int buttonState = 0;
int ledState = 0;

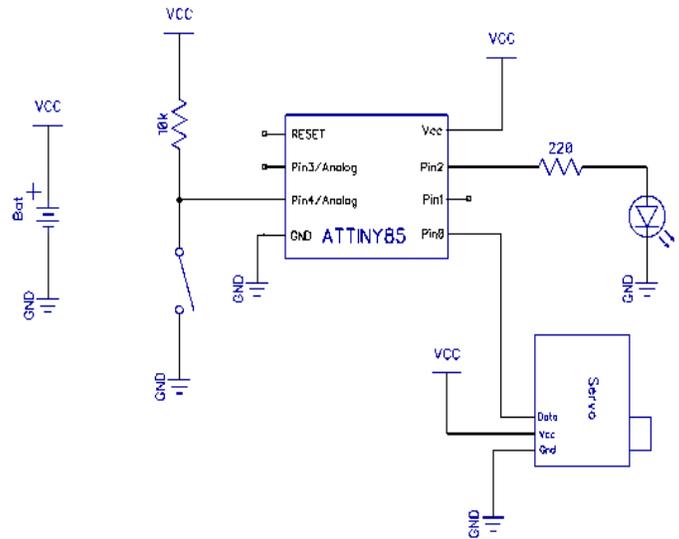
void setup() {
  pinMode(ButtonPin, INPUT);
  pinMode(LEDPin, OUTPUT);
  myServo.attach(9);
  Serial.begin(9600);
}

void loop() {
  buttonState = digitalRead(ButtonPin);
  Serial.println(buttonState);

  if (buttonState == 1) {
    digitalWrite(LEDPin, HIGH);
    myServo.write(90);
  }
}

```

4) Now that you have CODED the Arduino Nano and PROTOTYPED the circuit, you are going to make a schematic diagram of your circuit and a circuit board. The ONLY difference is that we are going to use an ATTINY85 microprocessor instead of the Arduino Nano as it is smaller and cheaper. The schematic to the right is the final result using the ATTINY85.

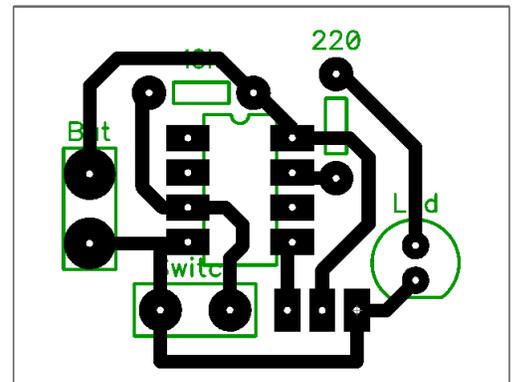


5) Open the program DIP TRACE SCHEMATIC on your computer. Once the program is open, go to the Electronics 9/10 website and under “Useless Box Project”;

- Open up the “[Using Diptrace – Library Setup](#)” YouTube video and complete the task.
- Now, open up the “[Making the Useless Box Schematic](#)” YouTube video and complete the task
- Your schematic in the DipTrace Schematic program will look identical to the one to the right once completed

6) Now you are going to design the Printed Circuit Board (PCB) using DIPTRACE PCB LAYOUT. You do not have to open it right now. Once the program is open, go to the Electronics 9/10 website and under “Useless Box Project”;

- Open up the “[Setting Up Diptrace PCB](#)” YouTube video and complete the task.
- Open up the “[Making the PCB Layout for the Useless Box](#)” YouTube video and complete the task.
- Once both Videos are complete, on your computer screen you should have a PCB Layout that looks pretty close to the one to the right.



7) Printing the circuit board. At this point you are either going to print out your own circuit board from the DIPTRACE program or get one from the teacher. If you print out your own, you have to ask the teacher for the special “Toner Transfer Paper” and load it into the printer with the clay side down. Then print to the “GPV-R500-CircuitBoard-P” printer.

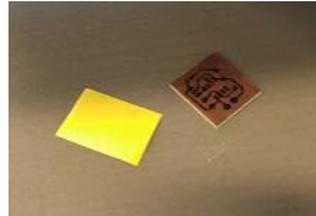
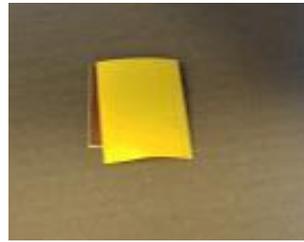
8) Once you have the circuit board printed on the special paper you will need to get a piece of “copper clad board” and cut it slightly bigger than the circuit board layout using the “sheet metal shear” from the metal shop.



9) Next, using steel wool, clean the surface of the copper clad board to remove any grease or dirt. After this process, try to not put your fingers on the copper.

10) You are going to “Transfer” the PCB layout from the paper to the copper using the “Toner Transfer Method”

- Turn on the T-shirt Press and let it heat up to 350 degrees. Also, ensure the timer is set to 20 seconds.
- Once the T-shirt Press has reached temperature, place the copper clad board in the center of the T-shirt Press with the PCB paper layout on top, with the black facing the copper.
- Press the lever down until it “clicks” in place. Wait until the timer beeps. “Pop” the lever up.
- **YOUR PROJECT IS HOT!! Use pliers to take the board off the bed.** Pull the paper off the copper clad board.
- Carefully check for “break” in the black “traces”. If the board is not perfect and there are “breaks”, ask for the special “circuit board pen” that can be used to touch up the spots. Also, put your INITIALS on the board so you can identify it!!



11) Etching your board. You are going to use the “Bubbler tank”. It has a chemical in it called “**Ammonium persulfate**” that eats copper away. Wherever the black from the toner is on the copper, the Ammonium Persulfate cannot get to and therefore leaves it. HOWEVER, leaving your board in the tank too long will result in ALL the copper being removed!! It should only be in the tank until the unwanted copper is removed.

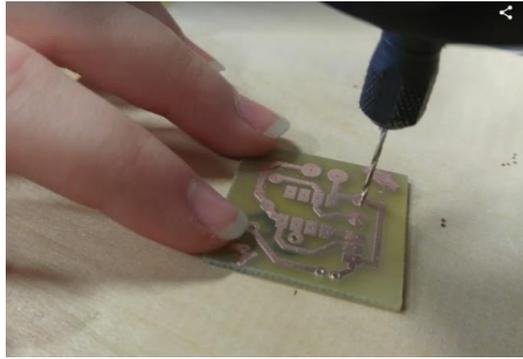
- First, drill a hole in one of the corners of your board using a dremel drill.
- Using a piece of “breadboarding wire” with the plastic coating on, thread it through the loop and make a loop. This will be used to hang your board in the tank.
- Go to the etching tank, open up the lid and hang your board. DO NOT DROP YOUR BOARD IN THE TANK ☹️
- Close the lid. When it is done, it will look like the image to the right 😊



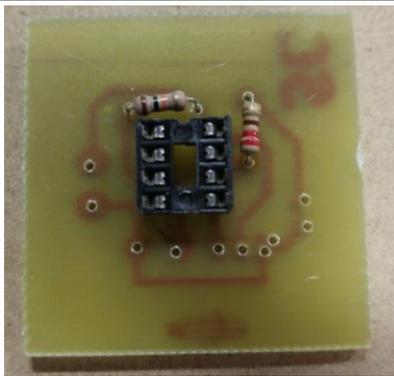
12) Remove the black toner transfer from the circuit board with steel wool.



- 13) Drill holes using the dremel drill.
 Drill holes in the centre of each pad.



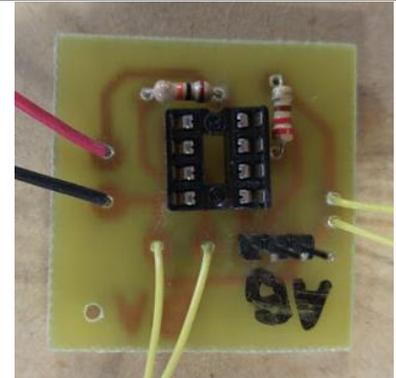
- 14) Place the components in place and solder as shown in the following pictures



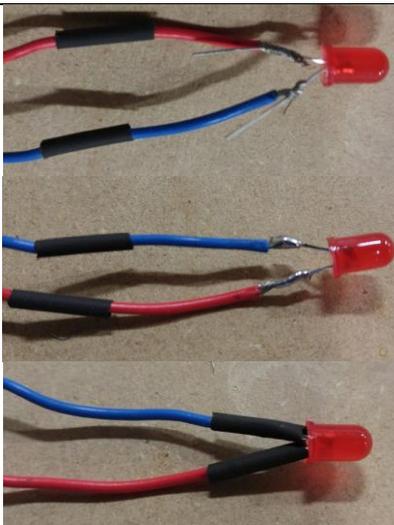
8 Pin DIP Socket
 10K Resistor
 220 Resistor



Solder all three components in place. Make sure the solders do not touch each other



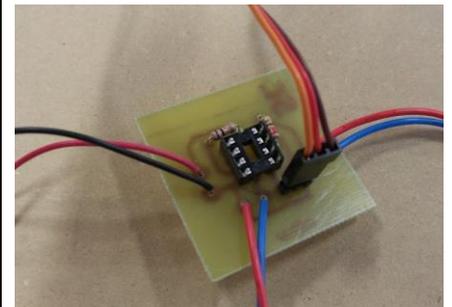
Insert the AA battery pack positive and negative wires as shown. DO NOT get these backwards. Also, insert and solder in the wires for the switch and LED



Add Heatshrink to the LED leads so that the wires cannot touch each other and short out. Notice the heat shrink is on the wires BEFORE soldering 😊

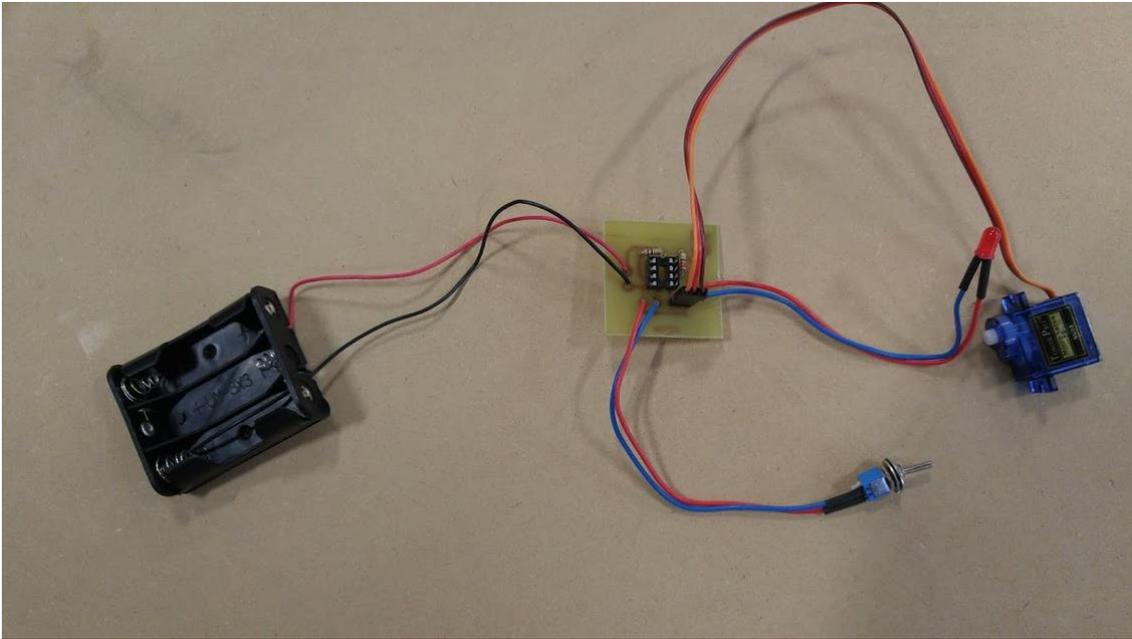


Solder wires to the switch. Notice the heat shrink on the wires first 😊 Make sure you do not burn out the switch by holding the solder pencil TOO LONG on the switch legs.



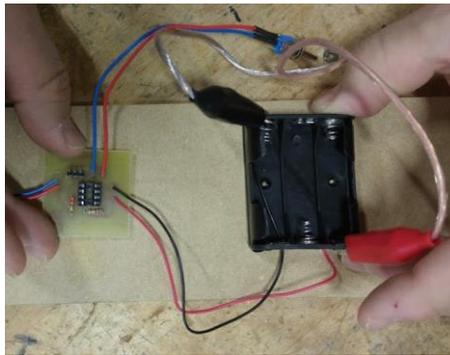
Insert the servo wire as shown. Notice the colours and what direction they go.

15) The finished product should look like the following. Wires can easily be shortened.....



Testing time!

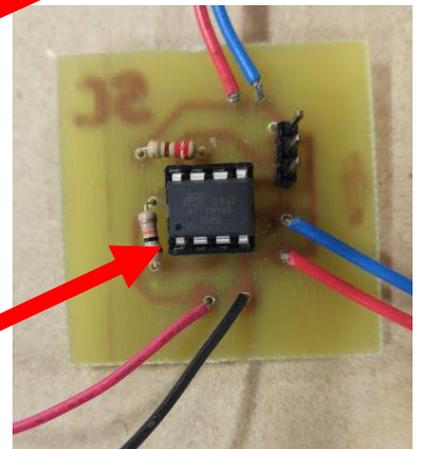
- FIRST - test the board for short circuits by attaching it to a power supply WITHOUT an ATTINY85 inserted into the 8 pin socket. Connect 5 volts to the battery as shown pack (get the positive and negative correct) and make sure the board does "short out" by looking at the power supply current needle. If the needle "pegs out" you have a problem. Did the servo twitch? If so, that is a GOOD thing!!



BAD!!!

- SECOND – disconnect the power and insert the ATTINY85 as shown. Note the direction. If you put it in the wrong way.....it will short out!!! Reconnect to the power supply that has been set to 5 volts. When you change the switch, does the LED turn on/off and does the servo move? If so, you are done and it work 😊 If not.....time to trouble shoot.

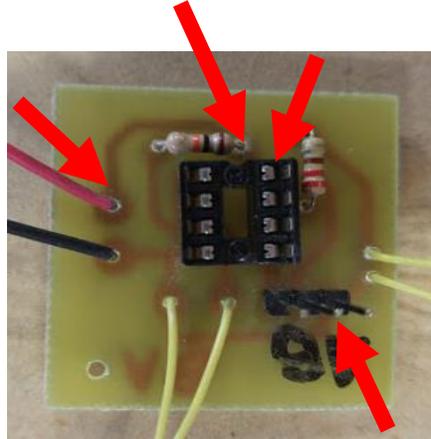
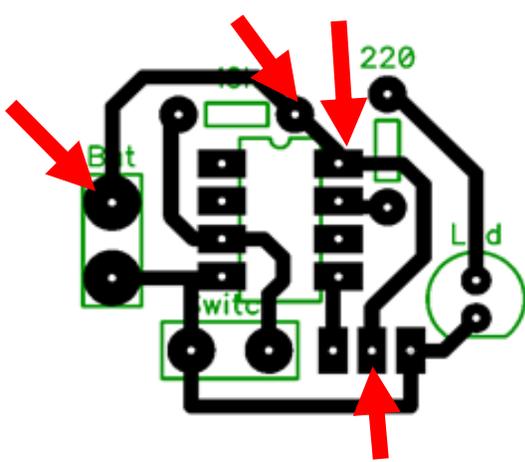
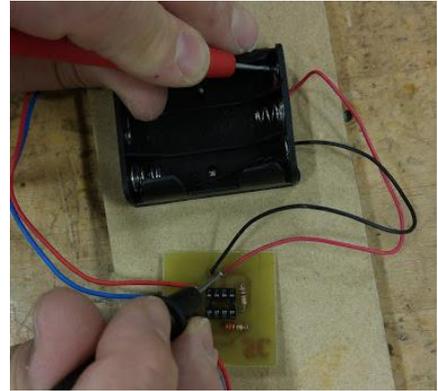
Dot



Troubleshooting Time!

STEP #1 (Establishing power)

So it doesn't work.....Start by testing for power. Using the "continuity" setting on the multi-meter, hold one of the test probes of the multi-meter on the "Positive" of the battery pack and then touch the other probe to the different spots on the board where "Positive" should be at. Does the multi-meter "BEEP"? If not, inspect the solder joints. If so, go to the next step.



Next test for "Negative" using the multi-meter. Move the probe to the "Negative" of the battery pack. Then test the points shown below. Does the multi-meter "BEEP"? If not, inspect the solder joints. If so, go to the next step.

