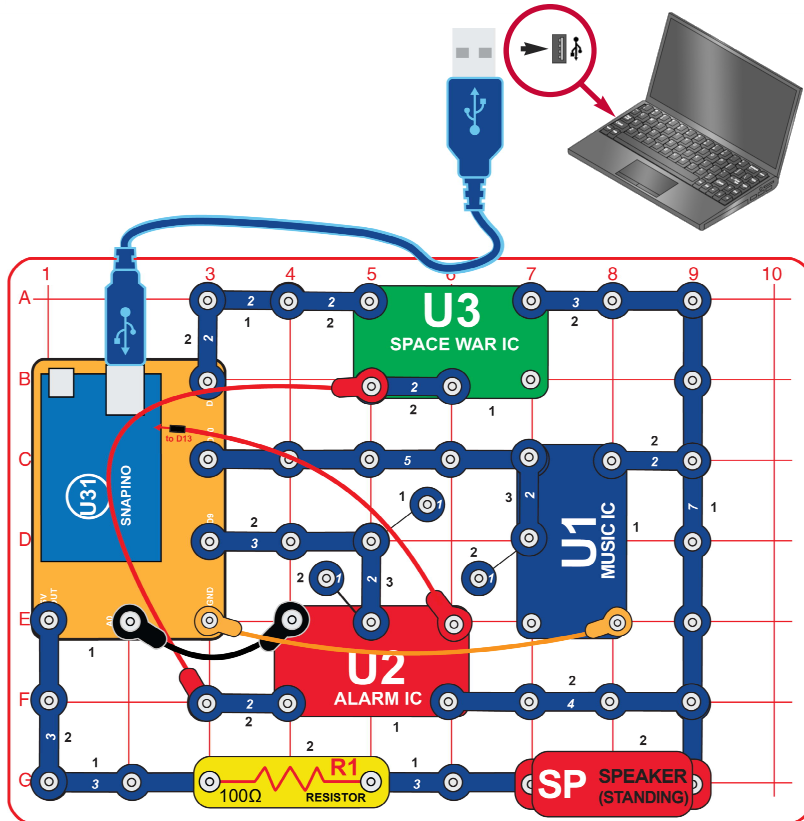




Project B1



Programming Note:

You can change how long each sound plays for by changing the delay value (now 4000, for 4 seconds) near the end of the sketch.

Multitune

Requires Model SC-300/500/750

This circuit requires parts from the SC-300, SC-500, or SC-750 sets. Build this circuit and load sketch Multitune into Snapino using the programming instructions in Snapino project 3. Snapino controls the music (U1), alarm (U2), and space war (U3) ICs, turning them on in order, but with 3 different sounds from the alarm IC.

```
Multitune | Arduino 1.6.9
File Edit Sketch Tools Help

Multitune
//Multitune

const int sound1 = 9;
const int sound2 = 10;
const int sound3 = 11;
const int sound4 = 12;
const int sound5 = A0;
int counter = 1;

void setup() {
  //Declare which variables are outputs or inputs
  pinMode(sound1,OUTPUT);
  pinMode(sound2,OUTPUT);
  pinMode(sound3,OUTPUT);
  pinMode(sound4,OUTPUT);
  pinMode(sound5,OUTPUT);
}

void loop() {
```



Project B2

Capacitor Monitor

Requires Model SC-300/500/750

This circuit requires parts from the SC-300, SC-500, or SC-750 sets. Build this circuit and load sketch Capacitor Monitor into Snapino using the programming instructions in Snapino project 3. When the upload is finished, click on the **Tools** menu, then pick **Serial Monitor** from the list. A window opens on your computer screen, displaying a graph. Snapino applies a voltage to a resistor-capacitor circuit (5.1K Ω resistor R3 and 470 μ F capacitor C5), then monitors and displays the voltage across the capacitor as it charges up. Then Arduino turns off the voltage to the resistor-capacitor circuit and monitors the capacitor voltage as it discharges.

The capacitor voltage is approximately 0V-5V, but is displayed as a number between 0 and 1023 (the value read by the analogRead command). Furthermore, when the value exceeds 1000 or gets below 10, the sketch reverses the charging/discharging.

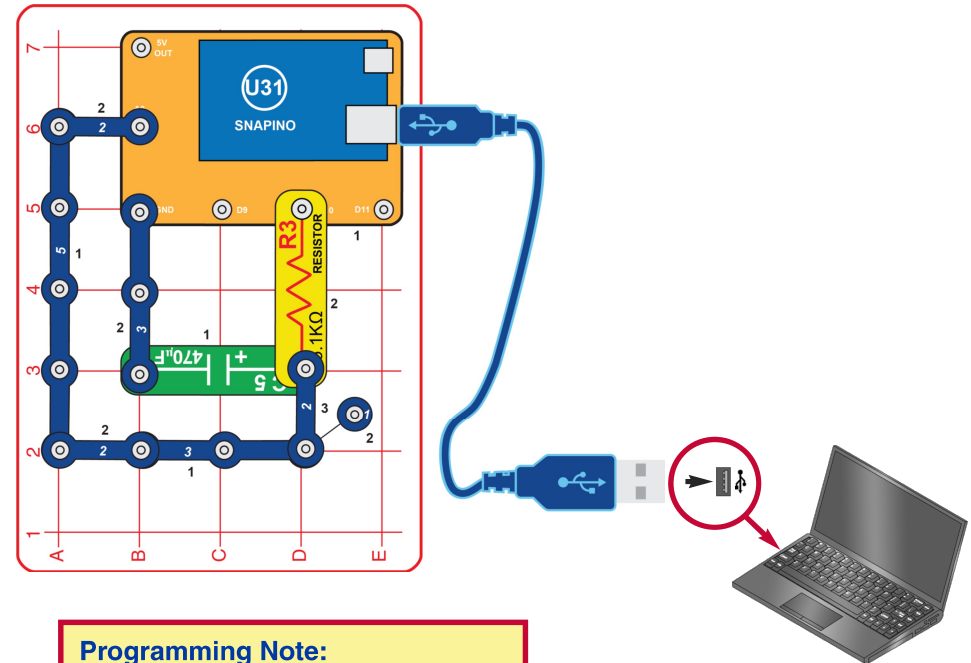
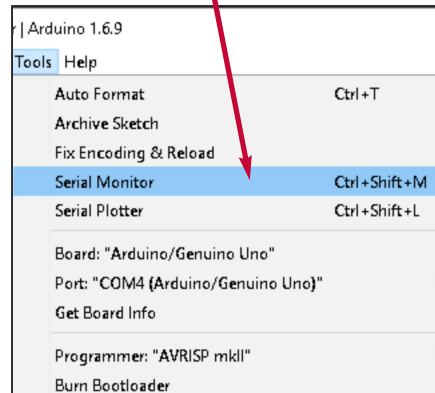
```
Capacitor_Monitor | Arduino 1.6.9
File Edit Sketch Tools Help

//Capacitor Monitor

const int output = 10;
int cval;
int val;
int timer;

void setup() {
  pinMode(output, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  cval = analogRead(0);
  Serial.println(cval);
  if(cval < 10){
    val = 1;
  }
  if(cval > 1000){
    val = 0;
  }
  if(val == 1){
    digitalWrite(output, HIGH);
  }
  if(val == 0){
    digitalWrite(output, LOW);
  }
  delay(100);
}
```

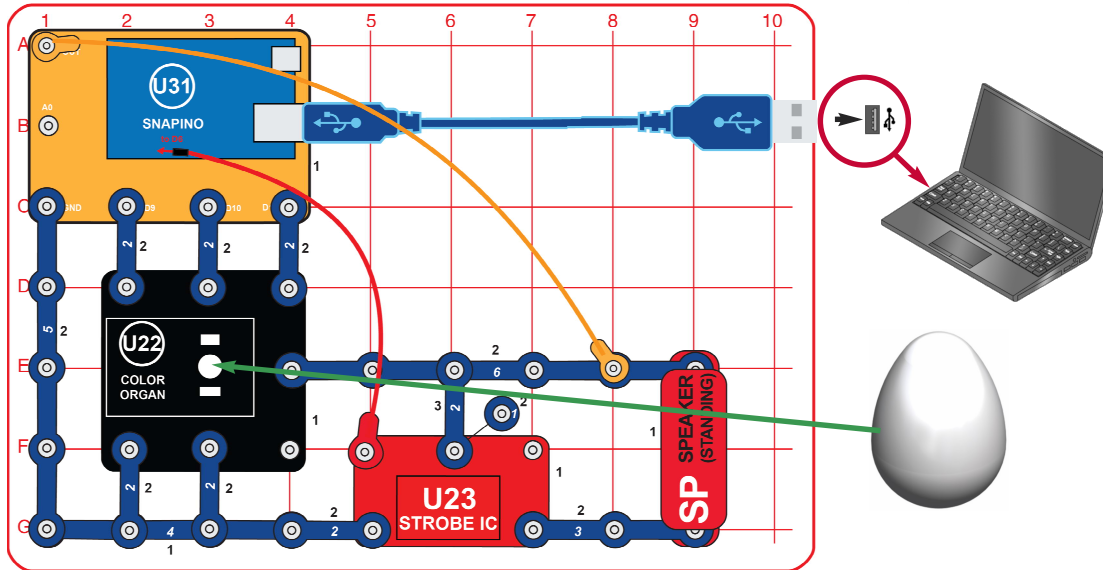


Programming Note:

Capacitors can store electricity. In this circuit the 470 μ F capacitor (C5) is charged or discharged by an electric current through the 5.1K Ω resistor (R3). The serial monitor displays the voltage across the capacitor. The rate of charging/discharging is determined by the values of R3 and C5; you can replace them with other ones you have and see how the charge/discharge rate changes.



Project B3



RGB Tones

Requires Model SCL-175 (Snap Circuits® Light)

This circuit requires parts from the SCL-175 Snap Circuits® Light set. Build this circuit and load sketch RGB Tones into Snapino using the programming instructions in Snapino project 3. Snapino controls the color organ (U22) and strobe IC (U23). The color organ displays red, green, and blue color, while the strobe IC makes a different tone sound for each color.

```
RGB_Tones | Arduino 1.6.9
File Edit Sketch Tools Help

//RGB Tones

const int strobe = 6;
const int blue = 11;
const int green = 10;
const int red = 9;
int counter = 1;

void setup() {
  pinMode(blue,OUTPUT);
  pinMode(green,OUTPUT);
  pinMode(red,OUTPUT);
  pinMode(strobe,OUTPUT);
}

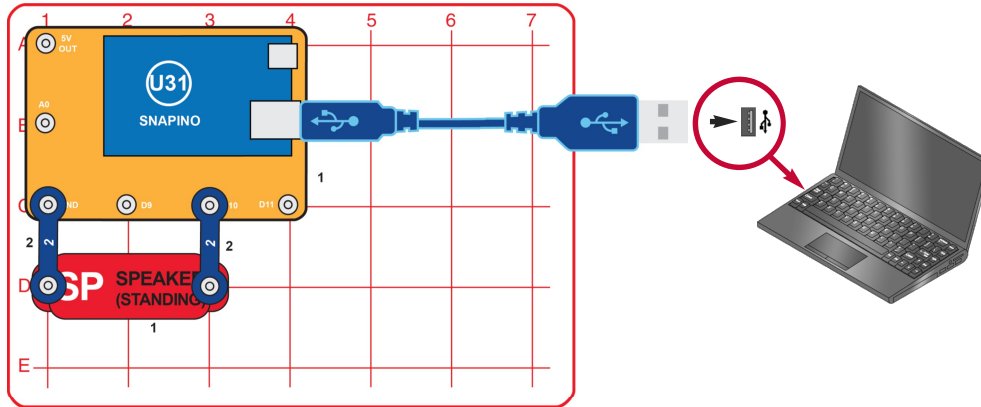
void loop() {

  if(counter == 1){
    analogWrite(strobe,900);
    digitalWrite(blue,HIGH);
    digitalWrite(green,LOW);
    digitalWrite(red,LOW);
  }
  if(counter == 2){
    analogWrite(strobe,300);
    digitalWrite(blue,LOW);
    digitalWrite(green,HIGH);
    digitalWrite(red,LOW);
  }
  if(counter == 3){
    analogWrite(strobe,30);
    digitalWrite(blue,LOW);
    digitalWrite(green,LOW);
    digitalWrite(red,HIGH);
  }
}
```



Project B4

Robotic Sounds



This circuit requires a speaker (SP, or SP2) from another Snap Circuits® set. Build this circuit and load sketch Robotic Sounds into Snapino using the programming instructions in Snapino project 3. Snapino plays robotic sounds on the speaker.

```
Robotic_Sounds | Arduino 1.6.9
File Edit Sketch Tools Help

Robotic_Sounds $
//Robotic Sounds

int sound;
void setup() {
}

void loop() {
  sound = random(31,1500); //Pick a random number between 31 and 1500.
  tone(10,sound);          //Output a tone at the frequency of the random number selected.
  delay(100);
}
```

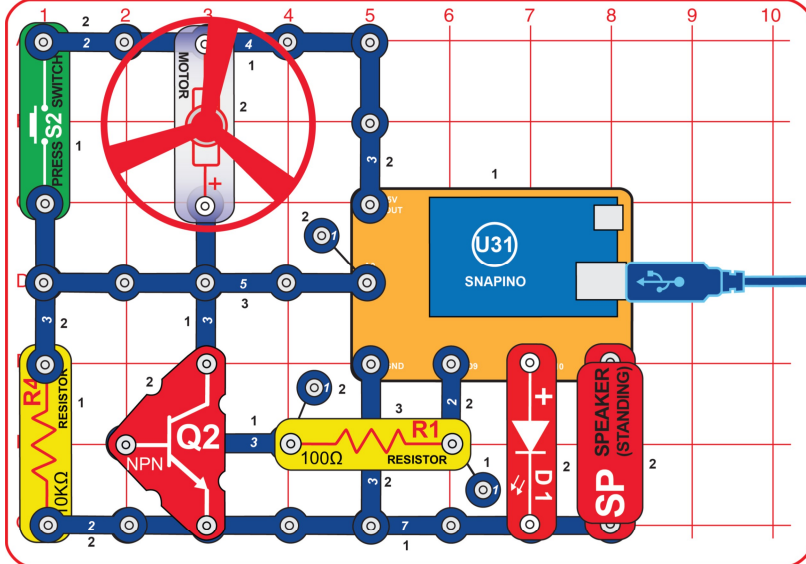
Programming Note: The random command picks a number between the 2 numbers shown (31 and 1500). The tone command outputs a tone at the frequency of the random number selected. You can change the sounds played by changing the minimum and maximum numbers for the random command, but the tone command requires a number between 31 and 65,535. This sketch uses a maximum of 1500, because your ears don't hear tones much higher than that.



Project B5

Countdown Fan Launcher

Requires Model SC-300/500/750



This circuit requires parts from the SC-300, SC-500, or SC-750 sets (SCL-175 can also be used using jumper wires in place of some snap wires). Build this circuit (note that there is a 3-snap wire under the NPN transistor (Q2)) and load sketch Countdown Fan Launcher into Snapino using the programming instructions in Snapino project 3. Push the press switch (S2) to start the launch sequence.

Note: if you replace the USB cable with the 9V battery connector after programming, use a new alkaline battery in it or the fan may not spin fast enough to fly (or may not spin at all). Replace the 9V battery if the circuit does not perform properly.



WARNING: Moving parts. Do not touch the fan or motor during operation. Do not lean over the motor.

```
Countdown_Fan_Launcher | Arduino 1.6.9
File Edit Sketch Tools Help

Countdown_Fan_Launcher
//Countdown Fan Launcher

const int LED = 10;
const int motor = 9;
const int bswitch = A0;
int button;
int oldbutton;
int buttonval;
int motorstate;
int speakerrate;
int startup;

void setup() {
  pinMode(LED, OUTPUT);
  pinMode(motor, OUTPUT);
  pinMode(bswitch, INPUT);
}

void loop() {
```



Project B6

Snapino Controlled House

Requires Model SC-3DMEG, SC-3Di, & SCJW-10

Parts List:

1-snap wire (6SC01) - 1
2-snap wire (6SC02) - 12
3-snap wire (6SC03) - 3
9V battery holder & switch (6SCB9) OR USB cable (9TLSCUSBAB) - 1
Full base grid (6SCBG) - 1
Mini base grid (6SCBGM) - 4
Mini base grid 2-sided (6SCBGM2) - 1
Base grid support (6SCBGSUP) - 4
White LED (6SCD6) - 1 (any LED can be substituted)
Color LED (6SCD8) - 1 (any LED can be substituted)
Blue LED (6SCD9) - 1 (any LED can be substituted)
Color2 LED (D12) - 1 (any LED can be substituted)
Black jumper wire (6SCJ1) - 1
Snap-to-pin wires (6SCJ5 RED or from SCJW10) - 3 total
Motor (6SCM4) - 1
Green fan (6SCM4F) - 1
Stabilizer (6SCSTAB) - 6
Light tunnel (6SCU30) - 1
Snapino module (6SCU31) - 1
Melody IC (6SCU32) - 1
Vertical snap wire (6SCV1) - 6
45 vertical snap wire (6SCV2) - 2

This circuit requires parts from SC-3DMEG, SC-3Di, & SCJW-10 sets.

Assembly (adult supervision is highly recommended):

1. Place base grid supports on base grid A.
2. Place parts on base grids B & C, and install into base grid supports on grid A. The pegs on grids B & C should be facing inward.
3. Place remaining parts on grid A.
4. Place the light tunnel (U30) on grid D, install grid D into the base grid support on grid A, finish connection of the V1 between grid D and grid C, and then add the remaining 2-snap wire on grid D.
5. Mount grids E & F (with pegs facing DOWN) at 45 degree angles on top of grids B & C, connecting the two V2 45 degree vertical snap wires and 6 stabilizers. Adjust the positions of the stabilizers as needed.
6. Install 1 snap-to-snap jumper wire and 3 snap-to-pin jumper wires. You may use any colors available.
7. Place wind fan on the motor (M4). Connect a USB cable to the Snapino module (U31) and program Snapino with sketch Snapino Controlled House. Once programed, you may replace the USB cable with 9V battery holder & switch.

The circuit will start once power is connected or programming is finished. The circuit will light the LEDs and light tunnel, spin the green fan, and play a tune on the melody IC.

```
Arduino IDE: Snapino_ControlledHouse | Arduino 1.8.5
File Edit Sketch Tools Help

Snapino_ControlledHouse

int wallLights = A0;
int doorbell = 9;
int fan = 11;
int red = 5;
int green = 6;
int blue = 4;

void setup() {
  pinMode(wallLights, OUTPUT);
  pinMode(doorbell, OUTPUT);
  pinMode(fan, OUTPUT);
  pinMode(red, OUTPUT);
  pinMode(green, OUTPUT);
}
```

