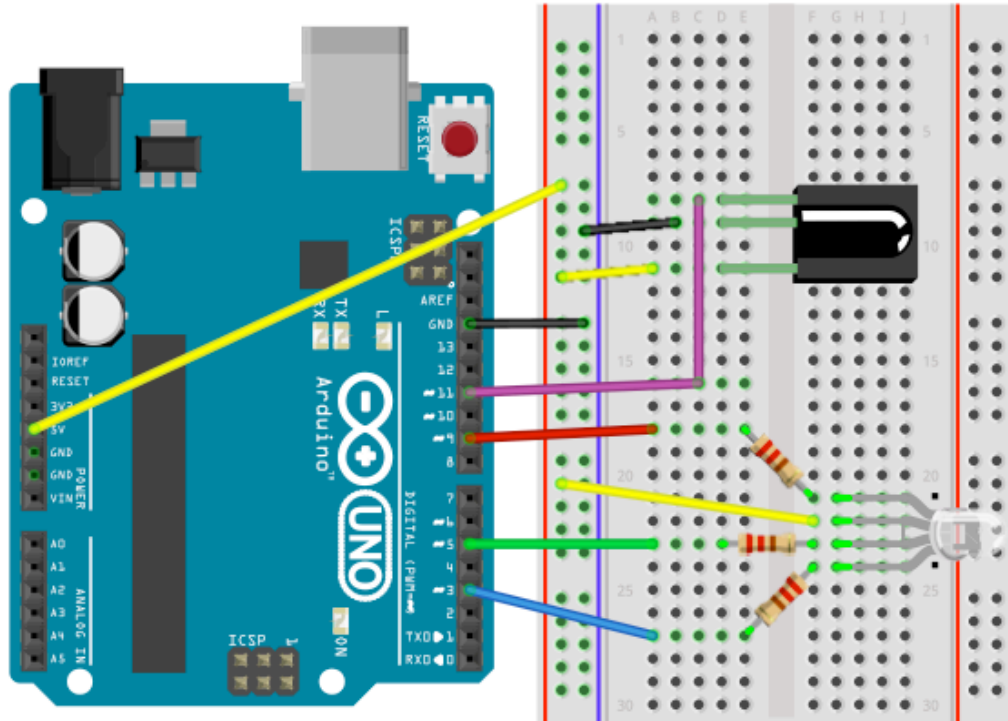


Activity #13: IR Communication, RGB LEDs and Speakers

CHALLENGE #1: Controlling an RGB LED with IR Communication

SETUP:



CODE:

```

/*
Adapted from IRremote: IRrecvDemo - demonstrates receiving IR codes with IRrecv.  An IR
detector/demodulator must be connected to the input RECV_PIN.
Version 0.1 July, 2009; Copyright 2009 Ken Shirriff (http://arcfn.com)
*/

```

```
#include <IRremote.h>
```

```
int ledState = 0;
int RECV_PIN = 11;
```

```
//RGB LED Pins
int ledRpin = 9;
int ledGpin = 5;
int ledBpin = 3;
```



```
IRrecv irrecv(RECV_PIN);

decode_results results;

//These numbers are too big to fit in "int"
unsigned long code0 = 4294967295;
unsigned long code1 = 2724022455;
unsigned long code2 = 2724048720;
unsigned long code3 = 2724046935;
unsigned long code4 = 2724014295;
unsigned long code5 = 2724004095;
unsigned long code6 = 2724037755;
unsigned long code7 = 2724039285;
unsigned long code8 = 2724049485;

void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn(); // Start the receiver
  pinMode(ledRpin, OUTPUT);
  pinMode(ledGpin, OUTPUT);
  pinMode(ledBpin, OUTPUT);
}

void loop() {
  if (irrecv.decode(&results)) {
    Serial.println(results.value, DEC);
    irrecv.resume(); // Receive the next value
  }

  //If a button is held down too long, ignore it.
  if(results.value == code0) { }

  //If power button is pressed, turn RGB LED on/off.
  else if (results.value == code1) {
    if (ledState == 0) {
      lightW();
      ledState = 1;
    }
    else {
      noLight();
      ledState = 0;
    }
    results.value = code0;
  }

  //All other buttons are assigned other actions.
  else if (results.value == code2) {
    cycleColors();
  }

  else if (results.value == code3) {
    lightR();
  }
}
```



```
}

else if (results.value == code4) {
  lightG();
}

else if (results.value == code5) {
  lightB();
}

else if (results.value == code6) {
  lightO();
}

else if (results.value == code7) {
  lightP();
}

else if (results.value == code8) {
  lightY();
}

}

void lightY () {
  analogWrite(ledRpin, 105);
  analogWrite(ledGpin, 240);
  analogWrite(ledBpin, 255);
}

void lightO () {
  analogWrite(ledRpin, 75);
  analogWrite(ledGpin, 248);
  analogWrite(ledBpin, 255);
}

void lightP () {
  analogWrite(ledRpin, 100);
  analogWrite(ledGpin, 255);
  analogWrite(ledBpin, 200);
}

void lightR () {
  analogWrite(ledRpin, 100);
  analogWrite(ledGpin, 255);
  analogWrite(ledBpin, 255);
}

void lightG () {
  analogWrite(ledRpin, 255);
  analogWrite(ledGpin, 200);
  analogWrite(ledBpin, 255);
}
```



PLT: Getting Started with Arduinos 2.0!

Michael Zitolo & Brian Palacios

```
void lightB () {  
  analogWrite(ledRpin, 255);  
  analogWrite(ledGpin, 255);  
  analogWrite(ledBpin, 100);  
}
```

```
void lightW() {  
  analogWrite(ledRpin, 40);  
  analogWrite(ledGpin, 120);  
  analogWrite(ledBpin, 253);  
}
```

```
void cycleColors() {  
  lightR();  
  delay(750);  
  lightO();  
  delay(750);  
  lightY();  
  delay(750);  
  lightG();  
  delay(750);  
  lightB();  
  delay(750);  
  lightP();  
  delay(750);  
  lightW();  
  delay(750);  
}
```

```
void noLight() {  
  analogWrite(ledRpin, 255);  
  analogWrite(ledGpin, 255);  
  analogWrite(ledBpin, 255);  
}
```

CHALLENGE #2: Controlling a Speaker using a Photo-resistor

Note: The potentiometer here serves only to control the volume of the speaker. It can be replaced with any resistor to permanently set the volume of the speaker.

CODE:

```
int sensorPin = 0;
int sensorValue = 0;
int speakerPin = 3;

//Range of Photo-resistor Values
int limit0 = 940;
int limit1 = 950;
int limit2 = 960;
int limit3 = 970;
int limit4 = 980;
int limit5 = 990;
int limit6 = 1000;

//Notes to Play
int noteA = 440;
int noteB = 494;
int noteC = 523;
int noteD = 587;
int noteE = 659;
int noteF = 698;
int noteG = 784;
int noteLength = 500; //Play notes for 500ms

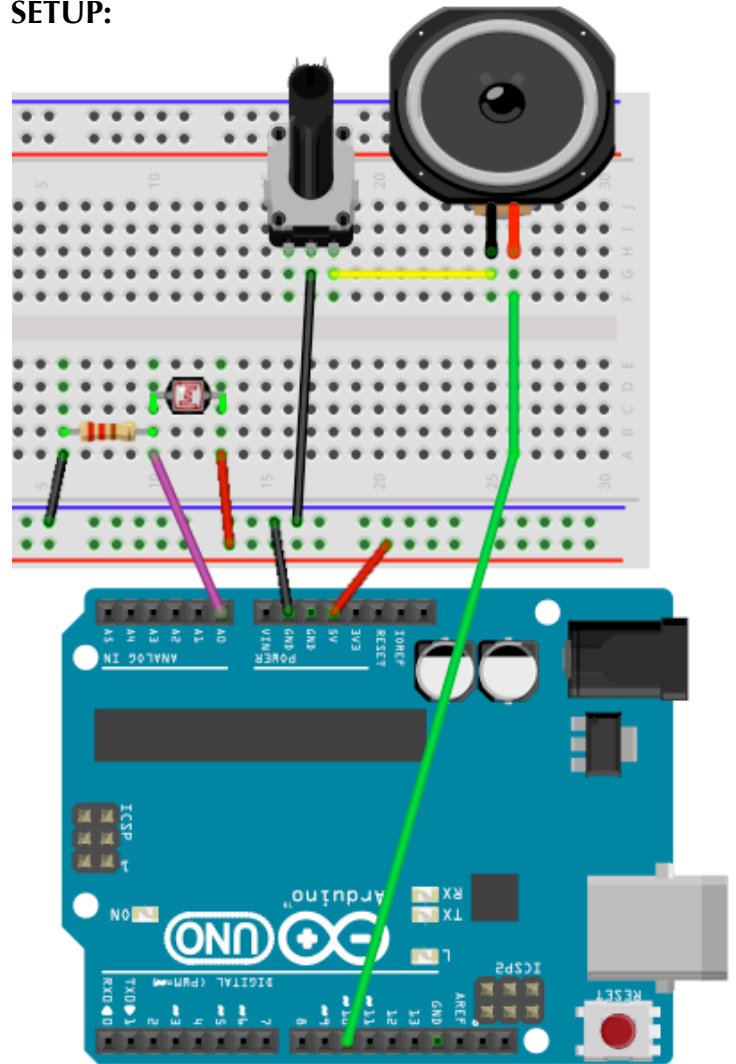
void setup(){
  Serial.begin(9600);
  pinMode(speakerPin, OUTPUT);
}

void loop(){
  delay(100);
  sensorValue = analogRead(sensorPin);
  Serial.println(sensorValue);

  if(sensorValue < limit0){
    //no tone
  }

  else if(sensorValue >= limit0 && sensorValue < limit1){
    tone(speakerPin, noteA, noteLength);
  }

  else if(sensorValue >= limit1 && sensorValue < limit2){
    tone(speakerPin, noteB, noteLength);
  }
}
```

SETUP:



PLT: Getting Started with Arduinos 2.0!

Michael Zitolo & Brian Palacios

```
else if(sensorValue >= limit2 && sensorValue < limit3){
  tone(speakerPin, noteC, noteLength);
}

else if(sensorValue >= limit3 && sensorValue < limit4){
  tone(speakerPin, noteD, noteLength);
}

else if(sensorValue >= limit4 && sensorValue < limit5){
  tone(speakerPin, noteE, noteLength);
}

else if(sensorValue >= limit5 && sensorValue < limit6){
  tone(speakerPin, noteF, noteLength);
}

else if(sensorValue >= limit6){
  tone(speakerPin, noteG, noteLength);
}

else {
  //no tone
}

}
```