How I Did It

Playing sounds using an Arduino Microcontroller, the VMusic2 MP3 playback module and a Radio Controller.

In this tutorial I'll be showing how I setup my first droid (R6-C9) to randomly play MP3's.

Things you will need:
1 Arduino Microcontroller. Any Arduino will work as long as it has at least one analog and two digital pins. In fact, almost any microcontroller will work, but my code was written for the Ardunio. To use another microcontroller will require writing your own code.

1 VMusic2.

1 R/C transmitter and receiver. For this I used the Cheap Control System's Cheap 6-Channel PS2 receiver, but a regular R/C system will work as well. They both send the same kind of signals to servos.

Wire. I used 22awg stranded wire to connect everything.

Wire Stripper.

1 standard servo. To test the R/C equipment.

I also used a screw-connector shield for the Arduino to simplify connecting everything to it and a VMusic2 connector kit. At this time I cannot remember where I purchased these two items or if they're even still available. Sorry.

Step One:
After you have everything you need, the first thing you need to do is measure the LOW and HIGH signal values coming from your R/C receiver. To do this you need to connect whichever R/C channel you want to use (I suggest a momentary toggle button) to one of the digital input pins on the Arduino.

A R/C channel on the receiver is a set of 3-pins, each labeled a different channel number (each channel corrasponds to a button, knob or stick on the R/C transmitter). The center pin is the voltage pin. We will not be using this pin for anything. The pins on each side are the ground and signal pins. Use a servo to determine which is which. Most standard R/C servo jumper wiring is arranged black/red/white. The ground is the black wire and the signal is the white wire (note: a few manufactures use different colors but the pin arrangement is the same), which leaves us with the red wire in the middle being the voltage wire (again, we will not be using the red wire).

Connect the signal wire from your chosen R/C channel to a digital input pin on the Arduino. Write down somewhere which pin you chose for future reference. Next, connect the ground wire from the same R/C channel to any GND pin on the Arduino. Now your remote control is connected to the Arduino. Don't forget to setup the power supply for both devices.

If you haven't already, install the Arduino sketching program on your computer so you can write and upload your "sketches" (or program code) to the Arduino. Once done, copy the following code into the program and then upload it to your Arduino.

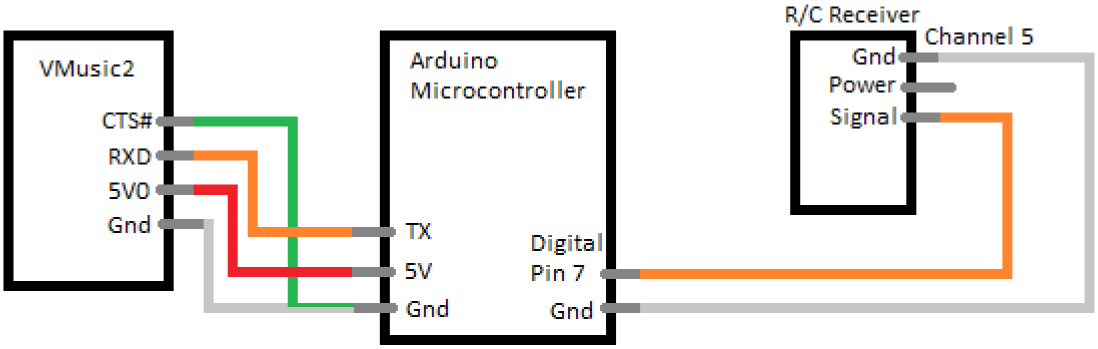
```
int rcPin = 7;    // select the input pin for the R\C receiver
int ledPin = 13;      // select the pin for the LED
int sensorValue = 0;  // variable to store the value coming from the sensor
unsigned long duration;

void setup() {
  Serial.begin(9600);         //  setup serial

  // declare the ledPin as an OUTPUT:
  pinMode(ledPin, OUTPUT);
  pinMode(rcPin, INPUT);
}

void loop() {
  duration = pulseIn(rcPin, HIGH);

  Serial.println(duration);           // debug value
}
```

Once this code is uploaded, open the serial monitor. You should see numbers scolling. Power on your R/C receiver and those numbers should start scrolling really fast. Now adjust the button/knob/stick you connected earlier and you should see the numbers change. What you need to write down is the number displayed when you are and are not activating the control on the transmitter. These two numbers will be your HIGH and LOW values you'll be using in the VMusic2 code.

Step Two:

Next you need to connect the VMusic2 to the Arduino. Click here for the VMusic2 data sheet showing the pin assignments, which I will be referencing.

First make sure the Arduino is powered off. Connect the VMusic2's pin 4 (RXD) to the TX pin on the Arduino, in my case, digital pin 1 (write down which digital pin you used for future reference). Next, connect pin 3 (5V0) to a 5V pin on the Arduino. This supplies the Vmusic2 with it's power. Next, connect both pins 1 (GND) and 6 (CTS#) to a GND pin on the Arduino. Don't worry if you need to connect multiple wires to the same ground pin. All ground pins are connected together so it doesn't matter. Congrats, your Vmusic2 is now connected to the Arduino. Power on the Arduino and ensure the Vmusic2 powers on as well. You should see a red light then green light on the Vmusic2. Steady green means ready.



Step Three:
Time to program. My sketch sample is made to randomly insert a number into the play command sent to the Vmusic2. To use it you need to adjust the pin assignments and signal LOW HIGH values in the sketch to match your setup. You'll then need to name your MP3 files 1 thru whatever and adjust the random number generator's number range to match the number of MP3's you have loaded onto the USB flash drive in the Vmusic2. To follow is how to make these adjustments.

First, here the Arduino sketch I'm currently using.

```
/* VMusic2 Random Sound Trigger
Author: John Flack
Version: APR1510
This program is designed to randomly select a number and
instruct the VMusic2 to play audio file with that number when a
button is activated on an R/C transmitter.
Special thanks to "Samurai" for Radio Control code examples.
*/

int RC5a = 7;  //gives name to pin 7, in this case, R/C ch5.
long trackN;  //sets variable holder for random in long character mode.

void setup()
{
 pinMode(RC5a, INPUT);  //sets pin 7 as an input.
 Serial.begin(9600);  //starts serial communication.
 randomSeed(analogRead(0));  //sets random to read noise from analog pin 0.
}
void loop()
{
 int RCval5 = pulseIn(RC5a, HIGH, 20000);  //read pin 7.
 delay(100);

 RCval5 = map(RCval5, 986, 1996, 1, 255);  //rcvr ch low/high converted to ASCII.

 trackN = random(1, 22);  //sets number range for random generator.

 if (RCval5 < 50 && RCval5 > 0)  //button pushed
 {
  Serial.print("VPF ");  //sends command to VM2.
  Serial.print(trackN);  //adds random number to VPF command.
  Serial.print(".mp3");  //added after random number.
   Serial.print(13, BYTE);  //required when sending a command to the VM2 for some reason.  won't work without.
 }
}
```

Ok. This sketch assumes your R/C receiver's channel is connected to digital pin 7 and uses the analog pin 0 for the random number generator.

The very first line of code, int RC5a = 7; names pin 7 RC5a. I named it this because I am using channel 5 on my R/C receiver. You will see RC5a in other lines of code. If your receiver is on a different pin #, say pin 5, you would change the 7 to a 5, int RC5a = 5;. You can also change the RC5a to anything you want, but you'll need to update it everywhere else it appears.

Line 16, RCval5 = map(RCval5, 986, 1996, 1, 255); sets the LOW and HIGH range of your R/C receiver's channel and also converts it to ASCII numbers. So in place of 986 put your LOW value. In place of 1996 put your HIGH value. Now the Arduino knows the signal range of your R/C channel.

Line 18, trackN = random(1, 22); is where you adjust the number range to be randomly selected. So if you have 30 MP3's, then you'd change the 22 to a 30. Easy.

You should now be ready to upload the sketch and test it. Some things to check before turning everything on:
Error check the sketch. It should be error free. If not, then there is a problem with the changes you made.
Make sure you upload the sketch to the Arduino.
The Vmusic2 cannot communicate to the Arduino if the Arduino is connected to your computer by USB. Unplug the USB cable and reset everything.
Make sure you have the MP3 file named as numbers, such as 9.mp3.
The Vmusic2 doesn't have a built-in amp for the speaker output. Ensure your speakers are connected to an amp or have one built-in them. I used a cheap set of desktop computer speakers.

If all goes well, you should get a randomly selected sound played whenevr you trigger the Arduino using your R/C transmitter. Enjoy.

John Flack

Reference Links:

Vmusic2 Data Sheet,
http://www.ftdichip.com/Support/Documents/DataSheets/Modules/DS_VMUSIC2.pdf

Vmusic2 Demo with Source Code at SurplusGizmos.com,
http://www.surplusgizmos.com/Vmusic2-Demo-with-Source-Code_ep_30.html

VMUSIC2 on Arduino at Tech Art Blog,
http://techartblog.blogspot.com/2007/08/vmusic2-on-arduino.html

Arduino Thread at Astromech.net,
http://astromech.net/forums/showthread.php?t=1035