

```

// Sheep Doorbell
// May 2011 by Todd Miller based on Adafruit example code
// Plays one sound when sensor is triggered, in this case sheep baaing Uses 7
sounds and picks one at random
// Parts include Arduino Uno, WAV Shield, small 8 ohm speaker and Sharp
GP2Y0A02 distance sensor
// To be mounted inside a Shaun The Sheep from Wallace & Gromet

#include <AF_Wave.h>
#include <avr/pgmspace.h>
#include "util.h"
#include "wave.h"
AF_Wave card;
File f;
Wavefile wave;    // only one!
uint8_t tracknum = 0;
uint16_t samplerate;

#define sensorIn 0
int songCount = 7;
int a[5];
int results;
int i=0;
int maxDist = 0;
int sensorRead = 0;
int fr = 0;
long previousMillis = 0;
long interval = 3000;

void setup() {
  Serial.begin(9600);          // set up Serial library at 9600 bps
  randomSeed(analogRead(1));
  Serial.println("Wave test!");
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(sensorIn, INPUT);
  if (!card.init_card()) {
    putstring_ni("Card init. failed!"); return;
  }
  if (!card.open_partition()) {
    putstring_ni("No partition!"); return;
  }
  if (!card.open_filesys()) {
    putstring_ni("Couldn't open filesystem"); return;
  }
}

```

```

}
if (!card.open_rootdir()) {
    putstring_ni("Couldn't open dir"); return;
}
putstring_ni("Files found:");
ls();

for (i=0; i<5; i++) { // Get max distance for comparison later
// maxDist += analogRead(5);
a[i]+=analogRead(sensorIn);
}
insertionSort(a,5);
ls();
}

void loop() {
    readSensor();
    uint8_t i, r;
    char c, name[15];
    unsigned long currentMillis = millis();

if(sensorRead > (maxDist +20) && (currentMillis - previousMillis > interval)){
    previousMillis = currentMillis;
    Serial.println(maxDist-25);
    int songNumber = random(songCount);
    Serial.print("Playing track # ");
    Serial.print(songNumber);
    Serial.print(" ");
    Serial.println(name);
    switch(songNumber)
    {
    case 0:
        playcomplete("BAA1.WAV");
        break;
    case 1:
        playcomplete("BAA2.WAV");
        break;
    case 2:
        playcomplete("BAA3.WAV");
        break;
    case 3:
        playcomplete("BAA4.WAV");
        break;
    case 4:
        playcomplete("BAA5.WAV");

```

```

break;
  case 5:
    playcomplete("BAA6.WAV");
    break;
  case 6:
    playcomplete("BAA7.WAV");
    break;
  }
  sensorRead=0;
}
card.reset_dir();
}

```

```

void playcomplete(char *name) {
  uint16_t potval;
  uint32_t newsamplerate;
  playfile(name);
  samplerate = wave.dwSamplesPerSec;
  while (wave.isplaying) {
    // you can do stuff here!
    delay(50);
  }
  card.close_file(f);
}

```

```

void playfile(char *name) {
  f = card.open_file(name);
  if (!f) {
    putstring_nl(" Couldn't open file"); return;
  }
  if (!wave.create(f)) {
    putstring_nl(" Not a valid WAV"); return;
  }
  // ok time to play!
  wave.play();
}

```

```

void readSensor(){
  for (int i=0; i < 5; i++){
    a[i]=analogRead(sensorIn);
  }
}

```

```

insertionSort(a,5);
Serial.print("max dis = ");
Serial.println(maxDist);
Serial.print("--- final sensor value = ");

```

```
Serial.print(sensorRead);  
Serial.println(" ---");  
}
```

```
void insertionSort(int arr[], int length) {  
    int i;  
    int j;  
    int tmp;  
    for (i = 1; i < length; i++) {  
        j = i;  
        while (j > 0 && arr[j - 1] > arr[j]) {  
            tmp = arr[j];  
            arr[j] = arr[j - 1];  
            arr[j - 1] = tmp;  
            j--;  
        }  
    }  
}
```

// array sorted, now lose low and high values for better average

```
a[0]=0;  
a[4]=0;  
for (i = 0; i < 5; i++){  
    results= results+a[i];  
}  
results/=3; // divide by 3 as places 1 and 5 = 0  
if (fr == 0)  
{  
    maxDist = results;  
    fr = 1;  
}  
else sensorRead=results;  
results=0;  
}
```

```
void ls() {  
    char name[13];  
    int ret;  
    card.reset_dir();  
    putstring_ni("Files found:");  
    while (1) {  
        ret = card.get_next_name_in_dir(name);  
        if (!ret) {  
            card.reset_dir();  
            return;  
        }  
        Serial.println(name);  
    }  
}
```

}