

Arduino Pinger Project

February 18, 2014

Text in sketch is color coded to what function it controls.

Color Key:

Pinger – Ultrasonic Range Sensor

Serial LCD Screen

Audio Player

LED Spots

LED Eyes

Servos

Standard Code

Notes

```
/*
 * Arduino Pinger Project
 *
 * Code for Milburn's Haunted Manor wall of bones and creatures scare.
 * The idea behind this sketch is to use the ultrasonic range finder to control
 * an event at three pre-set distances from the sensor.
 * The guests walk down a hall and come into the sensor range of the ultrasonic sensor.
 *
 * They get within the first increment range and set off the first series of events.
 * The critters start to move, a LED spot comes on, the eyes start to glow dimly.
 * Eerie sounds start coming out of the walls.
 * As the people progress down the hall this happens in two additional steps, the eyes glowing
 * brighter with each triggered step.
 * http://www.foolsquarter.com
 * http://www.milburnmanor.com
 * Tim Whitson - February 13, 2014
 * Please use and abuse this sketch as you like.
 */

#include <NewPing.h>
#include <SoftwareSerial.h>
#include <Servo.h>

#define txPin 2 // Arduino pin for the Rx connection on the serial LCD screen
#define PING_PIN 3 // Arduino pin tied to both trigger and echo pins on the ultrasonic sensor
#define MAX_DISTANCE 400 // Maximum sensor distance is rated at 400-500 cm

NewPing sonar(PING_PIN, PING_PIN, MAX_DISTANCE); // NewPing setup of pin and maximum distance.
SoftwareSerial LCD = SoftwareSerial(0, txPin);
```

```

const int ledspotPin1 = 4; // first led spot light
const int ledspotPin2 = 5; // second led spot light
const int ledspotPin3 = 7; // third led spot light
const int ledeyesPin1 = 6; // final all eyes on
const int audioPin1 = 8; // first audio file
const int audioPin2 = 12; // second audio file
const int audioPin3 = 13; // third audio file

const int LCDdelay=10; // conservative, 2 actually works

// Variables will change:

int ledspotState1 = LOW; // sets state of led spot light
int ledspotState2 = LOW;
int ledspotState3 = LOW;
int ledeyesState1 = LOW; // sets state of led eyes
int audioState1 = 1; // sets state of audio trigger
int audioState2 = 1;
int audioState3 = 1;
unsigned long interval=5000; // sets interval time between audio triggers, time set 180000 or 3 minutes.
unsigned long previousMillis=0;

Servo myservo1; // create servo object to control a servo
Servo myservo2;
Servo myservo3;

int potpin1 = 0; // analog pin used to connect the potentiometer
int potpin2 = 1;
int potpin3 = 2;
int val1; // variable to read the value from the analog pin
int val2;
int val3;

// wbp: goto with row & column
void lcdPosition(int row, int col) {
  LCD.write(0xFE); //command flag
  LCD.write((col + row*64 + 128)); //position
  delay(LCDdelay);
}
void clearLCD(){
  LCD.write(0xFE); //command flag
  LCD.write(0x01); //clear command
  delay(LCDdelay);
}
void backlightOn() { //turns on the backlight
  LCD.write(0x7C); //command flag for backlight stuff
  LCD.write(157); //light level
}

```

```

    delay(LCDdelay);
}
void backlightOff(){           //turns off the backlight
    LCD.write(0x7C);           //command flag for backlight stuff
    LCD.write(128);            //light level for off
    delay(LCDdelay);
}
void serCommand(){             //a general function to call the command flag for issuing all other commands
    LCD.write(0xFE);
}
void setup() {
    LCD.begin(9600);           // serial communications with LCD
    pinMode(txPin, OUTPUT);
    pinMode(ledspotPin1, OUTPUT);
    pinMode(ledspotPin2, OUTPUT);
    pinMode(ledspotPin3, OUTPUT);
    pinMode(ledeyesPin1, OUTPUT);
    pinMode(audioPin1, OUTPUT);
    pinMode(audioPin2, OUTPUT);
    pinMode(audioPin3, OUTPUT);
    backlightOn();             // turns on backligh of LCD
    clearLCD();                 // clears LCD screen
    lcdPosition(0,0);          // sets LCD position

    myservo1.attach(9);         // attaches the servo on pin to the servo object
    myservo2.attach(10);
    myservo3.attach(11);
}

void loop() {
    unsigned int uS = sonar.ping(); // Send ping, get ping time in microseconds (uS).
    clearLCD();
    lcdPosition(0,0);
    LCD.print("Ping: ");
    LCD.print(uS / US_ROUNDTRIP_CM); // Convert ping time to distance & print result (0 = outside set distance range, no
ping echo)
    LCD.print("cm");

    unsigned long currentMillis = millis(); // sets currentMillis to millis or time sketch has been running

    val1 = analogRead(potpin1);           // reads the value of the potentiometer (value between 0 and 1023)
    val2 = analogRead(potpin2);
    val3 = analogRead(potpin3);
    val1 = map(val1, 0, 1023, 0, 179);     // scale it to use it with the servo (value between 0 and 180)
    val2 = map(val2, 0, 1023, 0, 179);
    val3 = map(val3, 0, 1023, 0, 179);

    if (uS / US_ROUNDTRIP_CM <= 100)     // set this for distance to detect first event

```

```

{
digitalWrite(ledspotPin1, HIGH);
analogWrite(ledeyesPin1, 50); // starts to turn on eyes 50 of 255
myservo1.write(val1); // starts servo rotating at value set on pot1
if(audioState1 == 1){ // checks if time interval has passed to trigger audio file
  digitalWrite(audioPin1, HIGH);
  audioState1 = 0; // resets audio trigger so it won't trigger until interval is timed out
  previousMillis = currentMillis; // resets previousMillis to time when last audio triggered
}
}

if (uS / US_ROUNDTRIP_CM > 100)
{
digitalWrite(ledspotPin1, LOW);
analogWrite(ledeyesPin1, LOW); // only point where eyes should be turned off
myservo1.write(0); // stops servo rotation
}

if (uS / US_ROUNDTRIP_CM <= 75) // set this for distance to detect second event
{
digitalWrite(ledspotPin2, HIGH);
analogWrite(ledeyesPin1, 150);
myservo2.write(val2);
if(audioState2 == 1){
  digitalWrite(audioPin2, HIGH);
  audioState2 = 0;
}
}

if (uS / US_ROUNDTRIP_CM > 75)
{
digitalWrite(ledspotPin2, LOW);
myservo2.write(0);
}

if (uS / US_ROUNDTRIP_CM <= 50) // set this for distance to detect third event
{
digitalWrite(ledspotPin3, HIGH);
analogWrite(ledeyesPin1, 255);
myservo3.write(val3);
if(audioState3 == 1){
  digitalWrite(audioPin3, HIGH);
  audioState3 = 0;
}
}

if (uS / US_ROUNDTRIP_CM > 50)
{

```

```
digitalWrite(ledspotPin3, LOW);
myservo3.write(0);
}
if ((unsigned long)(currentMillis - previousMillis) >= interval) //checks to see if interval time has passed
{
  audioState1 = 1;           // resets audioState to allow audio to be triggered on next cycle
  audioState2 = 1;
  audioState3 = 1;
}

delay(250);                 // short delay
digitalWrite(audioPin1, LOW); // turns off audio trigger
digitalWrite(audioPin2, LOW);
digitalWrite(audioPin3, LOW);
}
```