

Project 20 - Ping Pong Code

Copy and Paste the code below into the Arduino sketch window. Verify and upload the code to the board.

```
// Include Libraries

#include "Arduino.h"

#include "NewPing.h"

#include "PiezoSpeaker.h"

#include "Button.h"

#include "Servo.h"

// Pin Definitions

#define HCSR04_PIN_TRIG    3

#define HCSR04_PIN_ECHO   4

#define PIEZOSPEAKER_PIN_SIG    5

#define PUSHBUTTON_PIN_1  6

#define SERVO9G_PIN_SIG    2

// Global variables and defines

unsigned int piezoSpeakerNNBBLength    = 6;

unsigned int piezoSpeakerNNBBMelody[]  = {NOTE_C4, NOTE_C4,
NOTE_A3, NOTE_D4, NOTE_C4, NOTE_A3};
```

```
unsigned int piezoSpeakerNNBBNoteDurations[] = {4, 4, 4, 4, 2, 4};
```

```
unsigned int piezoSpeakerArmedLength = 3;
```

```
unsigned int piezoSpeakerArmedMelody[] = {NOTE_C5, NOTE_C5, NOTE_C5};
```

```
unsigned int piezoSpeakerArmedNoteDurations[] = {4, 4, 4};
```

```
const int servo9gClosedPosition = 20; //Starting position - closed
```

```
const int servo9gOpenPosition = 110; //Trap door open position
```

```
const int triggerDistance = 10;
```

```
// Constructors
```

```
Servo servo9g;
```

```
NewPing hcsr04(HCSR04_PIN_TRIG,HCSR04_PIN_ECHO);
```

```
PiezoSpeaker piezoSpeaker(PIEZOSPEAKER_PIN_SIG);
```

```
Button pushButton(PUSHBUTTON_PIN_1);
```

```
// define vars for testing menu
```

```
#define TIMEOUT 10000 //define timeout of 10 sec
```

```
char menuOption = 0;
```

```
long time0;
```

```
/* This code sets up the essentials for your circuit to work. It runs first every  
time your circuit is powered with electricity. */
```

```
void setup() {
```

```
    // Setup Serial which is useful for debugging
```

```
    // Use the Serial Monitor to view printed messages
```

```
    Serial.begin(9600);
```

```
    while (!Serial) ; // wait for serial port to connect. Needed for native USB
```

```
    Serial.println("start");
```

```
    pushButton.init();
```

```
    servo9g.attach(SERVO9G_PIN_SIG);
```

```
    servo9g.write(servo9gClosedPosition);
```

```
    delay(500);
```

```
    servo9g.detach();
```

```
}
```

```
void loop() {

    bool triggered=false;

    bool armed=true;

    //Keep checking distance meter - if something close then open servo if not
    then back to wait

    while (triggered==false) {

        int hcsr04Dist = hcsr04.ping_cm();

        if((hcsr04Dist != 0) && (hcsr04Dist < triggerDistance)) {

            triggertrap(hcsr04Dist);

            triggered = true;

            armed = false;

        }

        else {

            //Serial.print(F("Distance: ")); Serial.print(hcsr04Dist);
            Serial.println(F("[cm]")); // print triggered distance for debug

            delay(100);

        }

    }

}
```

```
//Wait for reset signal
```

```
while (triggered==true) {
```

```
    bool pushButtonVal = pushButton.read();
```

```
    if(pushButtonVal) {
```

```
        resettrap();
```

```
        triggered = false;
```

```
    }
```

```
    else {
```

```
        delay(100);
```

```
    }
```

```
}
```

```
//Wait for second button press to signal that the trap is ready to arm again
```

```
while (armed==false) {
```

```
    bool pushButtonVal = pushButton.read();
```

```
if(pushButtonVal) {  
  
    armtrap();  
  
    armed = true;  
  
}  
  
else {  
  
    delay(100);  
  
}  
  
}  
  
}  
  
  
// This section is used to trigger the trap and play music  
  
void triggertrap(int triggerDistance)  
  
{  
  
    //Open servo trap  
  
    Serial.print(F("Trigger Distance: ")); Serial.print(triggerDistance);  
    Serial.println(F("[cm]")); // print triggered distance for debug  
  
    servo9g.attach(SERVO9G_PIN_SIG);    // 1. attach the servo to correct pin to  
    control it.  
  
    servo9g.write(servo9gOpenPosition); // 2. turns servo to open position.
```

```

delay(1000);           // 3. wait 100 milliseconds

servo9g.detach();     // 4. detach the servo

//Play Music- The Speaker will play the Nanny Nanny Boo Boo tune x2:P

piezoSpeaker.playMelody(piezoSpeakerNNBBLength,
piezoSpeakerNNBBMelody, piezoSpeakerNNBBNoteDurations);

delay(250);

piezoSpeaker.playMelody(piezoSpeakerNNBBLength,
piezoSpeakerNNBBMelody, piezoSpeakerNNBBNoteDurations);

delay(500);

}

void resettrap()

{

//Close servo trap

servo9g.attach(SERVO9G_PIN_SIG); // 1. attach the servo to correct pin to
control it.

servo9g.write(servo9gClosedPosition); // 2. turns servo to closed position.

delay(1000);           // 3. wait 100 milliseconds

servo9g.detach();     // 4. detach the servo

```

```
}
```

```
void armtrap()
```

```
{
```

```
    //Indicate trap armed
```

```
    piezoSpeaker.playMelody(piezoSpeakerArmedLength,  
piezoSpeakerArmedMelody, piezoSpeakerArmedNoteDurations);
```

```
    delay(500);
```

```
}
```