

```

1  //
2  // David Panamski - TUES - 8V
3  //
4  // Primary Idea based on:
https://www.electronicshub.org/arduino-joystick-interface-control-servo/
5  // Primary Idea based on: Arduino Joystick Interface - Control Servo using Arduino
   and Joystick
6  // Primary Idea based on: July 5, 2018, By Administrator
7  // Sound ISR based on: //code write by Moz for YouTube changel LogMaker360, 17-9-2015
8  // Sound ISR based on: Arduino KY-006 simple interrupt on a buzzer.
9  // Sound ISR based on: Code belongs to this video:
https://www.youtube.com/watch?v=cB1lc2RR4cQ
10
11  #include <Servo.h>
12
13  // Arduino MEGA2560 Hardware PIN Connection
14  const int led1GreenPin = 22;
15  const int led1RedPin = 23;
16  const int led2GreenPin = 24;
17  const int led2RedPin = 25;
18  const int buzzerPin = 26;
19  const int x_Analog_Joystick_Pin = A8;
20  const int y_Analog_Joystick_Pin = A9;
21  const int switch_Joystick_Pin = 18;
22  const int switch_Rotary_Pin = 19;
23  const int Servo_x_Control_Pin = 50;
24  const int Servo_y_Control_Pin = 51;
25
26  //General Constants
27  const int Time_Servo = 50;
28  const int Increment_Servo = 2;
29  const int X_pos_center = 90;
30  const int Y_pos_center = 90;
31
32  int xValue_Joystick;    //Analog Read from Joystick [0..1023]
33  int yValue_Joystick;    //Analog Read from Joystick [0..1023]
34  int X_pos = 90;         //Servo Angle Rotation [0..180] degree
35  int Y_pos = 90;         //Servo Angle Rotation [0..180] degree
36
37  int unsigned long button_time = 0;
38  int unsigned long last_button_time = 0;
39  int unsigned long button_time_Rotary = 0;
40  int unsigned long last_button_time_Rotary = 0;
41
42  volatile byte flag_Button_Toggle = 1;    //volatile is because its value can be
   changed somewhere
43  volatile byte flag_Button_Toggle_Rotary = 1;
44  volatile byte Increment_Servo_Multiplier = 1;
45  volatile byte flag_Sound_1 = 1;
46  volatile byte flag_Sound_2 = 0;
47  volatile byte flag_Mode = 1; // 0 - program mode // 1 - manual interactive mode
48
49
50  Servo Servo_X;
51  Servo Servo_Y;
52
53  void setup()
54  {
55    pinMode (x_Analog_Joystick_Pin, INPUT) ;
56    pinMode (y_Analog_Joystick_Pin, INPUT) ;
57    pinMode(led1GreenPin, OUTPUT);
58    pinMode(led1RedPin, OUTPUT);
59    pinMode(led2GreenPin, OUTPUT);
60    pinMode(led2RedPin, OUTPUT);
61    pinMode(buzzerPin, OUTPUT);
62
63    pinMode(switch_Joystick_Pin, INPUT_PULLUP);
64    pinMode(switch_Rotary_Pin, INPUT_PULLUP);
65    Servo_X.attach(Servo_x_Control_Pin );
66    Servo_Y.attach(Servo_y_Control_Pin);
67    Servo_X.write(X_pos_center);
68    Servo_Y.write(Y_pos_center);
69    Increment_Servo_Multiplier = 5;

```

```

70     digitalWrite(led1GreenPin, HIGH);
71     digitalWrite(led2GreenPin, LOW);
72     digitalWrite(buzzerPin, LOW);
73     Serial.begin(9600);
74     attachInterrupt(digitalPinToInterrupt(switch_Joystick_Pin), switchISR_Joystick,
CHANGE); //CHANGE //FALLING
75     attachInterrupt(digitalPinToInterrupt(switch_Rotary_Pin), switchISR_Rotary,
CHANGE); //CHANGE //FALLING
76 }
77
78 void loop()
79 {
80     sound();
81     digitalWrite(buzzerPin, LOW); // sound OFF
82     if(flag_Mode==1)    // 1 - manual interactive mode
83     {
84         manual_move();
85     }
86
87     if(flag_Mode==0)    // 0 - program mode
88     {
89         program_move();
90     }
91
92
93 }
94
95 void switchISR_Joystick()
96 {
97     button_time = millis();
98     if(button_time-last_button_time > 200)
99     {
100         if (flag_Button_Toggle==1) {
101             flag_Button_Toggle=0;
102             //Serial.println("Set Sound Off");
103             Increment_Servo_Multiplier = 1;
104             digitalWrite(led1GreenPin, LOW);
105             //digitalWrite(buzzerPin, LOW);
106             flag_Sound_1=1; //start while loop
107             // flag_Sound_1=0; //end while loop
108         }
109         else {
110             flag_Button_Toggle=1;
111             flag_Sound_1=1; //start while loop
112             //Serial.println("Set Sound On");
113             Increment_Servo_Multiplier = 5;
114             digitalWrite(led1GreenPin, HIGH);
115         }
116
117         last_button_time = button_time;
118     }
119
120 }
121
122 void switchISR_Rotary()
123 {
124     button_time_Rotary = millis();
125     if(button_time_Rotary-last_button_time_Rotary > 200)
126     {
127         if (flag_Button_Toggle_Rotary==1) {
128             flag_Button_Toggle_Rotary=0;
129             //Serial.println("switchISR_Rotary 0 - program mode");
130             flag_Mode = 0;    // 0 - program mode
131             digitalWrite(led2GreenPin, HIGH);
132             flag_Sound_2=1; //start while loop
133             // flag_Sound_1=0; //end while loop
134         }
135         else {
136             flag_Button_Toggle_Rotary=1;
137             flag_Sound_2=1; //start while loop
138             //Serial.println("switchISR_Rotary 1 - manual interactive mode");
139             flag_Mode = 1; // 1 - manual interactive mode
140             digitalWrite(led2GreenPin, LOW);

```

```

141     }
142
143     last_button_time_Rotary = button_time_Rotary;
144 }
145
146 }
147
148 void sound() {
149
150     while (flag_Sound_1==1)
151     {
152         for(int i=0; i<20; i++)
153         {
154             digitalWrite(buzzerPin, HIGH); //voice
155             delay (1); //delay 2ms
156             digitalWrite(buzzerPin, LOW); //no voice
157             delay (1); //delay 2ms
158         }
159         flag_Sound_1=0;
160     } //while Sound_1
161
162     while (flag_Sound_2==1)
163     {
164         for(int i=0; i<10; i++)
165         {
166             digitalWrite(buzzerPin, HIGH); //voice
167             delay (2); //delay 1ms
168             digitalWrite(buzzerPin, LOW); //no voice
169             delay (1); //delay 1ms
170         }
171         delay (50); //delay 20ms
172         for(int i=0; i<10; i++)
173         {
174             digitalWrite(buzzerPin, HIGH); //voice
175             delay (2); //delay 1ms
176             digitalWrite(buzzerPin, LOW); //no voice
177             delay (1); //delay 1ms
178         }
179         flag_Sound_2=0;
180     } //while Sound_2
181
182     digitalWrite(buzzerPin, LOW);
183
184 } // sound
185
186
187 void manual_move() {
188
189
190     xValue_Joystick = analogRead(x_Analog_Joystick_Pin);
191     yValue_Joystick = analogRead(y_Analog_Joystick_Pin);
192     // For X-Axis Servo
193     if (xValue_Joystick < 300)
194     {
195         if (X_pos < 10)
196         {
197             // Do Nothing
198         }
199         else
200         {
201             X_pos = X_pos - (Increment_Servo_Multiplier*Increment_Servo);
202             Servo_X.write(X_pos);
203             delay(Time_Servo);
204         }
205     }
206
207     if (xValue_Joystick > 700)
208     {
209         if (X_pos > 160)
210         {
211             // Do Nothing
212         }
213         else

```

```

214     {
215         X_pos = X_pos + (Increment_Servo_Multiplier*Increment_Servo);
216         Servo_X.write(X_pos);
217         delay(Time_Servo);
218     }
219 }
220 // For Y-Axis Servo
221 if (yValue_Joystick < 300)
222 {
223     if (Y_pos < 10)
224     {
225         // Do Nothing
226     }
227     else
228     {
229         Y_pos = Y_pos - (Increment_Servo_Multiplier*Increment_Servo);
230         Servo_Y.write(Y_pos);
231         delay(Time_Servo);
232     }
233 }
234
235 if (yValue_Joystick > 700)
236 {
237     if (Y_pos > 160)
238     {
239         // Do Nothing
240     }
241     else
242     {
243         Y_pos = Y_pos + (Increment_Servo_Multiplier*Increment_Servo);
244         Servo_Y.write(Y_pos);
245         delay(Time_Servo);
246     }
247 }
248
249 } // manual_move
250
251 void program_move() {
252     int i,j;
253
254     sound();
255     Servo_X.write(0);
256     Servo_Y.write(0);
257     for(i=0;i<=180;i++){
258         Servo_X.write(i);
259         for(j=0;j<=180;j++){
260             Servo_Y.write(j);
261             delay(Time_Servo/5);
262             if(flag_Mode==1) return; // 1 - manual interactive mode
263         }
264         //delay(Time_Servo/5);
265     }
266 }
267
268 }
269

```