***Dancing of servo motor***

#include <Servo.h>

#define NUM\_SERVOS 32

Servo myServo[NUM\_SERVOS];

void setup()

{

// Attach pins from the Arduino Mega board to the Servo objects.

// Starting from pin 22, there happen to be exactly 32 pins on the double row pins.

for( int i=0; i<NUM\_SERVOS; i++)

{

myServo[i].attach( i + 22); // pin 22 up to 53 is 32 pins

}

}

void loop()

{

// Sequence one.

// All servo motor are set to a random angle.

for( int a=0; a<15; a++)

{

for( int i=0; i<NUM\_SERVOS; i++)

{

myServo[i].write( random( 0, 181));

delay( 2);

}

delay( 150);

}

// Sequence two.

// All servo motors move with the same angle.

for( int i=0; i<NUM\_SERVOS; i++)

{

myServo[i].write( 0); // set to begin position (horn is rotated left)

}

delay( 1000); // wait to let the viewer get used to it

for( int a=0; a<3; a++)

{

for( int r=0; r<=180; r++) // move horns to the right

{

for( int i=0; i<NUM\_SERVOS; i++)

{

myServo[i].write( r);

}

delay( 6);

}

for( int r=180; r>=0; r--)

{

for( int i=0; i<NUM\_SERVOS; i++) // move horns to the left

{

myServo[i].write( r);

}

delay( 6);

}

}

// Sequence three.

// A rotating wave.

for( int a=0; a<6; a++)

{

for( int i=0; i<NUM\_SERVOS; i++)

{

for( int j=0; j<NUM\_SERVOS; j++)

{

// Calculate distance to active servo

int d = j - i;

if( d < 0)

d = -d;

if( d > (NUM\_SERVOS / 2))

d = NUM\_SERVOS - d;

int angle = 90 - (10 \* d);

if( angle < 0)

angle = 0;

myServo[j].write( angle);

}

delay(40);

}

}

// Sequence four.

// A "compass"

// Start by pointing upwards

int pointer = NUM\_SERVOS \* 3 / 4;

showPointer( pointer);

delay( 1000); // let the viewer get used to new pattern

for( int i=0; i<5; i++)

{

showPointer( --pointer);

delay( 150);

}

delay( 200);

for( int i=0; i<9; i++)

{

showPointer( ++pointer);

delay( 150);

}

delay( 200);

for( int i=0; i<5; i++)

{

showPointer( --pointer);

delay( 150);

}

delay( 200);

for( int i=0; i<4; i++)

{

showPointer( ++pointer);

delay( 150);

}

delay( 160);

for( int i=0; i<2; i++)

{

showPointer( --pointer);

delay( 150);

}

delay( 80);

for( int i=0; i<1; i++)

{

showPointer( ++pointer);

delay( 150);

}

delay( 2000);

}

// This function makes a "pointer" with the servos.

// It is used to create the "compass".

// The parameter 's' is the servo motor that has the pointer.

// It is allowed that 's' is below zero or larger than the numbers of servo motors.

void showPointer( int s)

{

int pointerA = s % NUM\_SERVOS; // Using the '%' (remainder) for valid number

int pointerB = (s + 1) % NUM\_SERVOS; // pointer is made with the next servo motor

int tailA = (s + 16) % NUM\_SERVOS;

int tailB = (s + 17) % NUM\_SERVOS;

// make pointer with servo motor s and s+1.

myServo[pointerA].write(180-56);

myServo[pointerB].write(56);

// make tail with servo motor s+16 and s+17.

myServo[tailA].write(95);

myServo[tailB].write(85);

// Set servos right of pointer

int n = (NUM\_SERVOS / 2) - 2;

int start = pointerB + 1;

for( int i=0; i<n; i++)

{

int j = (start + i) % NUM\_SERVOS;

myServo[j].write( 2);

}

// Set servos left of pointer

start = tailB + 1;

for( int i=0; i<n; i++)

{

int j = (start + i) % NUM\_SERVOS;

myServo[j].write( 178);

}

}

// The function GenerateDiagram() can be used to generate

// the diagram.json file for Wokwi.

// To use it, call it from the setup() function, and the

// serial output can be copied into the diagram.json file.

void GenerateDiagram()

{

Serial.begin(115200);

Serial.print( "{\n");

Serial.print( " \"version\": 1,\n");

Serial.print( " \"author\": \"Generated\",\n");

Serial.print( " \"editor\": \"wokwi\",\n");

Serial.print( " \"parts\": [\n");

Serial.print( " {\n");

Serial.print( " \"type\": \"wokwi-arduino-mega\",\n");

Serial.print( " \"id\": \"mega\",\n");

Serial.print( " \"top\": 270,\n");

Serial.print( " \"left\": 185,\n");

Serial.print( " \"attrs\": {}\n");

Serial.print( " },\n");

// Put the servo motor in reverse order in the diagram.json

// I think that is visually better.

// The horn now overlaps the next servo when the horn moves to the right.

for( int i=NUM\_SERVOS-1; i>=0; i--)

{

float rotate = float( i) \* (360.0 / float( NUM\_SERVOS));

float rad = rotate / 360.0 \* 2.0 \* M\_PI;

float top = (300.0 \* sin( rad)) + 300.0;

float left = (300.0 \* cos( rad)) + 300.0;

Serial.print( " {\n");

Serial.print( " \"type\": \"wokwi-servo\",\n");

Serial.print( " \"id\": \"servo");

Serial.print( i);

Serial.print( "\",\n");

Serial.print( " \"top\": ");

Serial.print( top);

Serial.print( ",\n");

Serial.print( " \"left\": ");

Serial.print( left);

Serial.print( ",\n");

Serial.print( " \"rotate\": ");

Serial.print( rotate);

Serial.print( ",\n");

Serial.print( " \"attrs\": { \"hornColor\": \"Red\" }\n");

Serial.print( " }");

if( i != 0)

Serial.print( ",");

Serial.print( "\n");

}

Serial.print( " ],\n");

Serial.print( " \"connections\": [\n");

for( int i=0; i<NUM\_SERVOS; i++)

{

int j = i + 1;

if( j == NUM\_SERVOS)

j = 0;

Serial.print( " [ \"servo");

Serial.print( i);

Serial.print( ":V+\", \"servo");

Serial.print( j);

Serial.print( ":V+\", \"Red\", [] ],\n");

Serial.print( " [ \"servo");

Serial.print( i);

Serial.print( ":GND\", \"servo");

Serial.print( j);

Serial.print( ":GND\", \"Black\", [] ],\n");

Serial.print( " [ \"mega:");

Serial.print( i + 22);

Serial.print( "\", \"servo");

Serial.print( i);

Serial.print( ":PWM\", \"Green\", [ ] ],\n");

}

Serial.print( " [ \"mega:GND.2\", \"servo9:GND\", \"Black\", [ ] ],\n");

Serial.print( " [ \"mega:5V\", \"servo9:V+\", \"Red\", [ ] ]\n");

Serial.print( " ]\n");

Serial.print( "}\n");

}