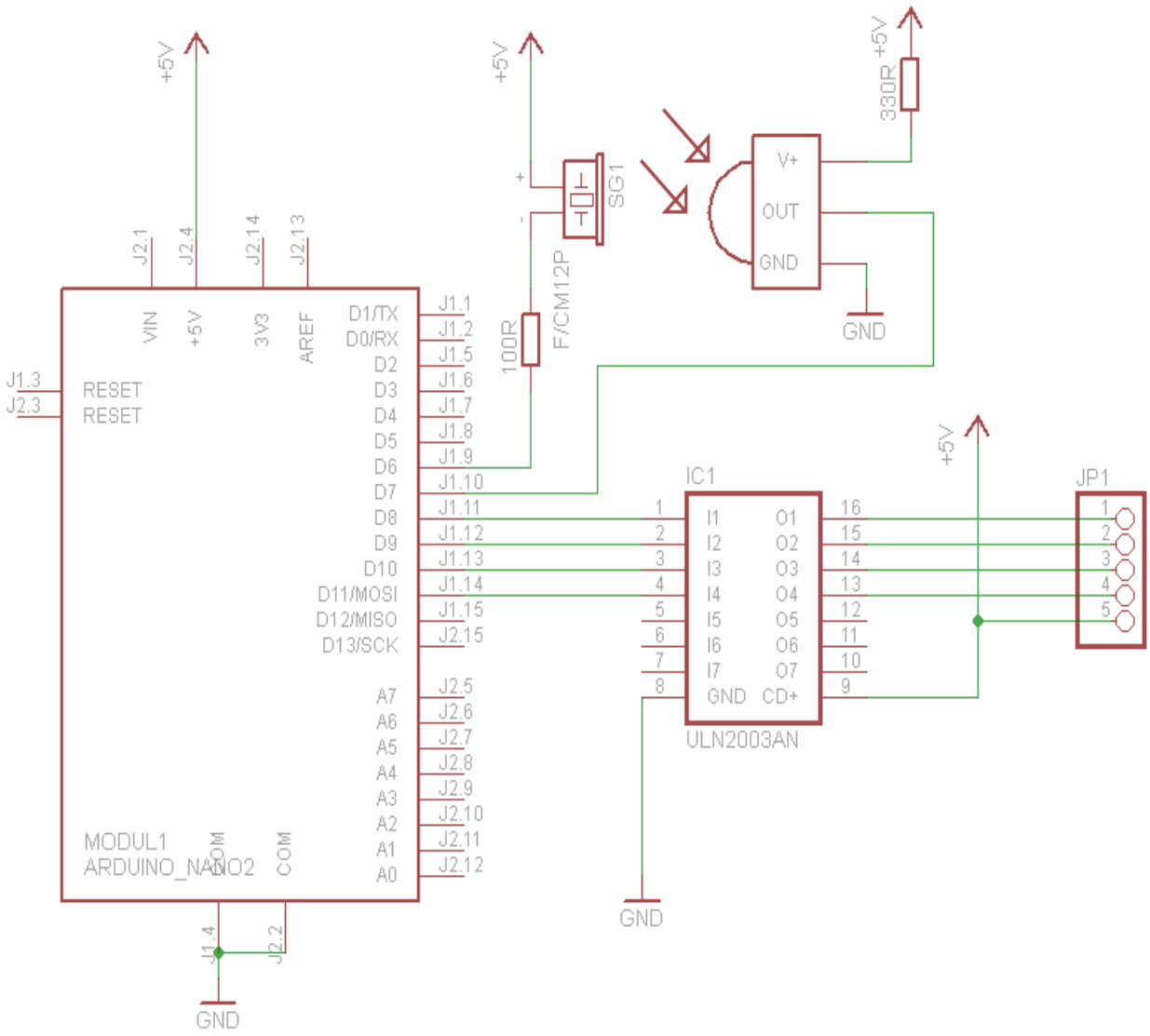
Rotary Table for 3D scanning.

IR Remote Controlled Rotary Table for 3D scanning.

To build it is needed:
12 pcs steel ball 1/4"
Screw M4 / 25 mm
Bearing 4x9x4 mm
Arduino Nano
Stepper motor 28BYJ-48
Driver ULN2003
Buzzer
IR detector
IR remote control
Power connector



RotaryTable.ino file:

```
#include "Stepper.h"
#include "IRremote.h"

/* Rotary Table of own construction
 *
 * There is a pinion with 20 teeth on the motor axis
 * There are 150 teeth along the circumference of the plate,
 * the ratio is 1:7.5
 * One turn of the motorcycle is 2048 steps,
 * the whole turn of the table is 15360 steps
 */

/*----- Variables, Pins -----*/
#define STEPS1   2048 // Number of steps per revolution of Internal shaft
#define STEPS2  15360 // Number of steps per revolution of Table
const int RECV_PIN = 7; // Signal Pin of IR receiver to Arduino Digital Pin
7
const int BUZZ_PIN = 6; // Signal Pin out buzzer Digital Pin 6
int  IsInput;     // 1 - keyboard input
int  StepsInput;  // number from keyboard
int  Steps2Take;  // number steps confirmed

/*-----( Declare objects )-----*/
// Setup of proper sequencing for Motor Driver Pins
// In1, In2, In3, In4 in the sequence 1-3-2-4

Stepper stepper(STEPS1, 8, 10, 9, 11);
IRrecv irrecv(RECV_PIN);    // create instance of 'irrecv'
decode_results results;     // create instance of 'decode_results'

/*-----------------------------------------------------------------------*/

void playTone(int tone, int duration) {
  for (long i = 0; i < duration * 1000L; i += tone * 2) {
    digitalWrite(BUZZ_PIN, HIGH);
    delayMicroseconds(tone);
    digitalWrite(BUZZ_PIN, LOW);
    delayMicroseconds(tone);
  }
}

void playNote(char note, int duration) {
  char names[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C' };
  int tones[] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014, 956 };

  // play the tone corresponding to the note name
  for (int i = 0; i < 8; i++) {
    if (names[i] == note) {
      playTone(tones[i], duration);
    }
  }
}

void turnOff(int tone, int duration) {
                     digitalWrite(8,LOW);
                     digitalWrite(9,LOW);
                     digitalWrite(10,LOW);
                     digitalWrite(11,LOW);
```

```
}

void addInput(int key) {
  StepsInput = 10*StepsInput + key;
}

/*********************************************************************/

void setup()
{
  IsInput = 0;
  Steps2Take = 2560;   // do  -- 60°;
  Serial.begin(9600);  // open a serial connection to your computer
  irrecv.enableIRIn(); // Start the receiver
  irrecv.blink13(true);
  pinMode(BUZZ_PIN, OUTPUT); // Set buzzer - pin as an output
}

void loop()
{
if (irrecv.decode(&results)) // have we received an IR signal?
  {
  Serial.println(results.value, HEX);


    switch(results.value)

    {
                //   break;
      case 0xD7E84B1B: //Press "OK" Turn off Motor
                    turnOff;
                    break;

      case 0xFF906F: // Right button pressed
                     playNote('g', 10);
                     stepper.setSpeed(7.5); // 1 rpm
                     stepper.step(-Steps2Take); //
                     delay(100);
                     turnOff;
                     break;

     case 0xFFE01F: // Left button pressed case 0xA3C8EDDB:
                     playNote('g', 10);
                     stepper.setSpeed(7.5); // 1 rpm
                     stepper.step(Steps2Take); //
                     delay(100);
                     turnOff;
                     break;

     case 0xFFE21D: // Menu button pressed
                     playNote('c', 40);
                     delay(100);
                     playNote('C', 40);
                     IsInput = 1;
                     StepsInput = 0;
                     delay(100);
                     break;

     case 0xFFC23D: // Return button pressed
                     if (IsInput = 1)
                     {
```

```
                playNote('C', 40);
                delay(100);
                playNote('C', 40);
                if (StepsInput == 0)
                {
                 StepsInput = 1;
                }
                Steps2Take = STEPS2 / StepsInput;
                if (Steps2Take == 0)
                {
                  Steps2Take = 1;
                }
               }
               IsInput = 0;
               break;


case 0xFFB04F: // Clear button pressed
               playNote('C', 40);
               delay(100);
               playNote('c', 40);
               IsInput = 0;
               break;

case 0xFF6897: // 0 button pressed
               if (IsInput == 1)
               {
                 playNote('c', 40);
                 delay(100);
                 addInput(0);
               }
               break;

case 0xFF30CF: // 1 button pressed
               if (IsInput == 1)
               {
                 playNote('c', 40);
                 delay(100);
                 addInput(1);
               }
               break;

case 0xFF18E7: // 2 button pressed
               if (IsInput == 1)
               {
                 playNote('c', 40);
                 delay(100);
                 addInput(2);
               }
               break;

case 0xFF7A85: // 3 button pressed
               if (IsInput == 1)
               {
                 playNote('c', 40);
                 delay(100);
                 addInput(3);
               }
               break;

case 0xFF10EF: // 4 button pressed
```

```
                    if (IsInput == 1)
                    {
                      playNote('c', 40);
                      delay(100);
                      addInput(4);
                    }
                    break;

      case 0xFF38C7: // 5 button pressed
                    if (IsInput == 1)
                    {
                      playNote('c', 40);
                      delay(100);
                      addInput(5);
                    }
                    break;

      case 0xFF5AA5: // 6 button pressed
                    if (IsInput == 1)
                    {
                      playNote('c', 40);
                      delay(100);
                      addInput(6);
                    }
                    break;

      case 0xFF42BD: // 7 button pressed
                    if (IsInput == 1)
                    {
                      playNote('c', 40);
                      delay(100);
                      addInput(7);
                    }
                    break;

      case 0xFF4AB5: // 8 button pressed
                    if (IsInput == 1)
                    {
                      playNote('c', 40);
                      delay(100);
                      addInput(8);
                    }
                    break;

      case 0xFF52AD: // 9 button pressed
                    if (IsInput == 1)
                    {
                      playNote('c', 40);
                      delay(100);
                      addInput(9);
                    }
                    break;
      break;

     }
       irrecv.resume(); // receive the next value
   }

}/* --end main loop -- */
```

Method of use:

First set the number of steps per table turn. For example, 30 steps, this corresponds to 12° one step. From the controller, press the keys:

MENU
3
0
RETURN

Now the **LEFT** / **RIGHT** keys rotate the table one step at a time.

Comment:

To cancel the setting operation, press the **C** key.

The number of steps should be any multiple of 1 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 3 * 5, for example: 1, 2, 3, 4, 5, 6, 8, 10, 12, 15, 16, 20, 24, 30, 32, 40, 48, 60, 64, 80, 96, 120....