Instituto Tecnológico de Estudios Superiores Monterrey, Campus Monterrey
Applied Robotics
Sergio W. Sedas Gersey
Final Project: Piano Robot

The Piano Robot

David Alejandro Kutugata Gastélum - A01138865
Franco Eduardo Trujillo Ramón - A0809188
Roberto De la Garza Alvídrez - A00812611

Introduction:

One of the most amazing things in life is music, a particular way humans express themselves by creating or reproducing melodies by playing many kinds of instruments. The piano, a classic and noble instrument, is one of the commonly used for this purpose, we take advantage of our five-fingered hands to play at full capacity.

The question is: can a robot play a piano as well as a human? There are several kinds of robots, each with different types of motors, sensors, actuators, everyone with its own advantage...like humans. Our project proposal is a robot which is able to play a piano or keyboard, being custom built, controlled with an Arduino microcontroller and will include the usage of sensors and motors kinematics for a proper functionality.

The three levels of accomplishment are:

-Minimum: the robot will press keys of the keyboard with at least one finger and create melodies, like a regular person who wants to learn piano but has no guidance from a teacher.

-Target: the robot will have five fully functional “fingers” that will press keys and create melodies. The robot will be able to move and play all notes within at least two octaves of the keyboard.

-Wow: the robot will work a s described on Target and will feature a playlist of melodies to play on command from the computer of the user. The communication between the robot and the computer will be wireless. The robot will be able to perform a piece from German composer Ludwig van Beethoven.

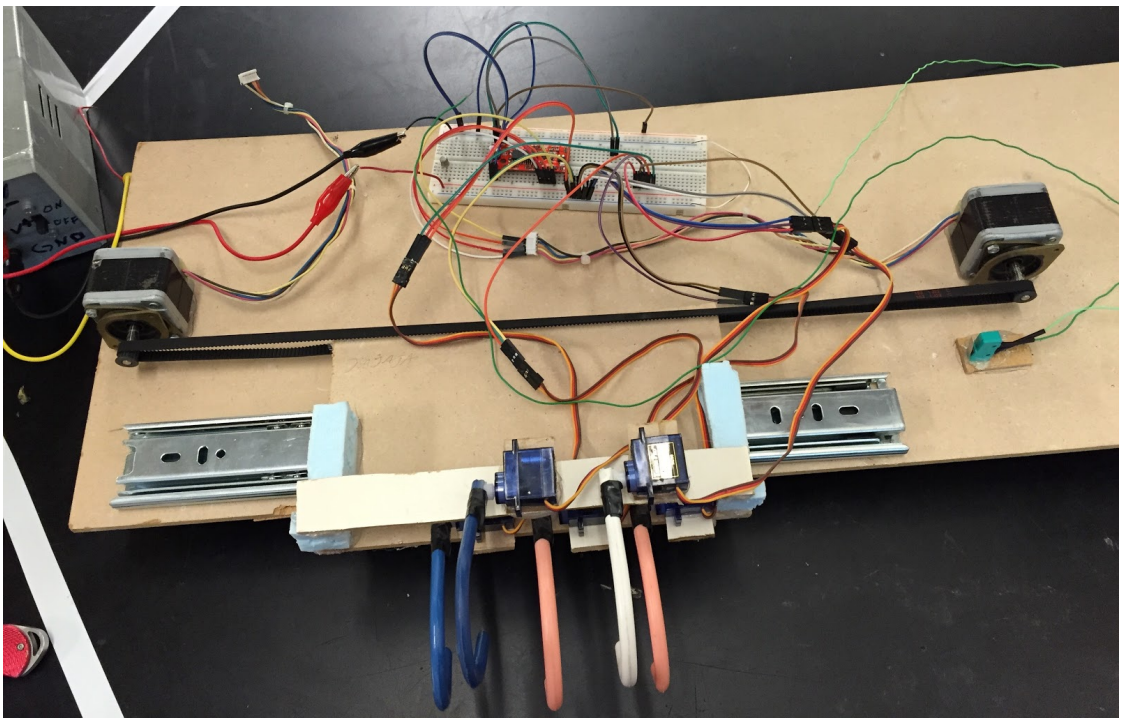
Assembling and hardware design:

For the robot, we decide for a cartesian type with 5 fingers simulated by RC-servos. In order to move in one axis we use a stepper motor that moves a belt that is attached to the robot fingers and for friction reduce and support we used a drawer’s slider.

- -Materials and tools
    - Mechanical
        - Belt
        - Slider
        - Wood
    - Electronics/ Electromechanic
        - EasyDriver
        - Arduino UNO
        - Stepper motor
        - Limit switch
        - RC-Servos
        - Piano

- Software
  - Arduino IDE
  - ServoLibrary
  - StepperLibrary

Final Design:



Program and logic:

The functionality of the robot was inspired in the working of a normal printer, calibrating itself and setting its ‘0’ position and making its own world reference to locate each tone in the piano.
For the tempo, depending of the song we defined a normal temp in 4/4, also for some songs and because of our robot limitations we made some fixes to the original tablatures to made easy for our robot to play them.

Issues and solutions

During the assembling we have several issues, one of them was the motor fix on the right place for a straight and tense band, it only need a little effort and patience to solve this. The other problem was the usage of the driver for the step-motor manipulation, we were trying to make it work. We saw some tutorials and datasheet and we accomplish the pulse manipulations.

Also, the velocity manipulation for the band was a problem, we saw another tutorials to see how can we make it go faster. Then we find it out by changing a velocity and acceleration parameter in the driver code. The fingers is still a problem to make them stay

on place. We used hooks for fingers simulation, but we pasted them at the servo's rotation axis, but the forces that deals with separates them after a couple of songs.

Another issue with the band was the implementation of a switch as an indicator for a limit, we did´´ use a pull-up resistor and the band move without limits, crashing with the steppers. We solve this adding the resistor. Almost finishing with our problems, the design of the table wasn't as precise as we wanted, because of the materials and size, but we adapted the project to it.

Finally the codification and coordination for the playlist, we changed servos in code, force of pushing and translational movements, but finally after a lot of work and dedication we were able to code the songs.

Final Conclusions:

The most important lesson we got from this robot is that we are capable of achieving. At first the idea sounds fun, but ridiculous. There is no purpose to have a robot play an instrument other than entertainment. But if we can build a custom robot to play a piece from Beethoven, we can surely build another one to assemble a car, or to make surgery. Given more time and less other subjects to deal with, the robot would have had a bigger playlist, but for now, it is capable of playing three pieces. Finally, we can surely say that our project was successful.

Annex:

Code:

```
#include <Servo.h>
#define DIR_PIN A5
#define STEP_PIN 8
int tono=-1150;
int semi=-575;
Servo m1;
Servo m2;
Servo m3;
Servo m4;
Servo m5;
bool limit=true;

void setup()
{
  m1.attach(2);
  m2.attach(3);
  m3.attach(4);
  m4.attach(5);
  m5.attach(6);
```

```
  m1.write(45);
  m2.write(45);
  m3.write(45);
  m4.write(45);
  m5.write(45);
  pinMode(DIR_PIN, OUTPUT);
  pinMode(STEP_PIN, OUTPUT);
  pinMode(7,INPUT_PULLUP);
}

void loop()
{
  findZero();
    iron();
  findZero();
  stay();
  findZero();
  Beethoven();

}

void stay()
{
  rotate(4*tono,1);
  m1.write(0);
  delay(1500);
  m1.write(45);
  m2.write(17);
  delay(1000);
  m2.write(45);
  m1.write(0);
  delay(1500);
   m1.write(45);
   delay(50);
   m1.write(0);
   delay(1000);
   m1.write(45);
   m2.write(17);
   delay(1000);
   m1.write(0);
   m2.write(45);
   delay(1500);
   m1.write(45);
   //parte doremi
   rotate(3*tono,1);
   m1.write(0);
```

```
delay(250);
m1.write(45);
m2.write(17);
delay(250);
m3.write(0);
m2.write(45);
delay(250);
 m1.write(0);
m3.write(45);

delay(250);
m1.write(45);
m2.write(17);
delay(250);
m3.write(0);
m2.write(45);
delay(250);
 m1.write(0);
m3.write(45);

delay(250);
m1.write(45);
m2.write(17);
delay(250);
m3.write(0);
m2.write(45);
delay(250);
 m1.write(0);
m3.write(45);
delay(200);
    m1.write(45);
    delay(50);
    m1.write(0);
    delay(500);
    m1.write(45);
//
delay(250);
   m1.write(0);
delay(250);
m1.write(45);
m2.write(17);

delay(250);
m3.write(0);
m2.write(45);
delay(250);
```

```
    m1.write(0);
   m3.write(45);

   delay(250);
   m1.write(45);
   m2.write(17);
   delay(250);
   m3.write(0);
   m2.write(45);
   delay(250);
    m1.write(0);
   m3.write(45);

   delay(250);
   m1.write(45);
   m2.write(17);
   delay(250);
   m3.write(0);
   m2.write(45);
   delay(250);
   m3.write(45);
   m5.write(0);
   delay(250);
       m5.write(45);
       m3.write(0);
       delay(1000);
       //
       m1.write(0);
       m3.write(0);
       m5.write(0);
       delay(2000);
       m1.write(45);
       m3.write(45);
       m5.write(45);



}

void iron()
{
  int i=3;
  rotate(5*tono,.5);
  while(i>0)
  {
      m1.write(0);
```

```
delay(1000);
m1.write(45);
delay(50);
rotate(2*tono,1);
m1.write(0);
delay(1000);
m1.write(45);
delay(50);
m1.write(0);
delay(500);
m1.write(45);
m2.write(17);
delay(500);
m2.write(45);
delay(50);
m2.write(17);
delay(1000);
m2.write(45);
m4.write(17);
delay(250);
m4.write(45);
m3.write(0);
delay(250);
m3.write(45);
m4.write(17);
delay(250);
m4.write(45);
m3.write(0);
delay(250);
m3.write(45);
m4.write(17);
delay(250);
m4.write(45);
m3.write(0);
delay(250);
m3.write(45);
m1.write(0);
delay(500);
m1.write(45);
delay(50);
m1.write(0);
delay(500);
m1.write(45);
m2.write(17);
delay(500);
m2.write(45);
```

```
      delay(50);
      m2.write(17);
      delay(500);
      m2.write(45);
      rotate(-2*tono,1);
      i--;
  }
}

void Beethoven()
{
  m3.write(0);
  delay(500);
  m3.write(45);
  delay(50);
  m3.write(0);
  delay(500);
  m3.write(45);
  m4.write(17);
  delay(500);
  m4.write(45);
  m5.write(10);
  delay(500);
  m5.write(25);
  delay(100);
  m5.write(10);
  delay(500);
  m5.write(25);
  //primer compas
  m4.write(17);
  delay(500);
  m4.write(45);
  m3.write(0);
  delay(500);
  m3.write(45);
  m2.write(17);
  delay(500);
  m2.write(45);
  //segundo compas
  m1.write(0);
  delay(500);
  m1.write(45);
  delay(50);
  m1.write(0);
  delay(500);
  m1.write(45);
```

```
  m2.write(17);
  delay(500);
  m2.write(45);
  m3.write(0);
  delay(500);
  m3.write(45);
  //tercer compas
  delay(50);
  m3.write(0);
  delay(800);

  m2.write(17);
  delay(250);
  m2.write(45);
  delay(250);
  m2.write(17);
  delay(1000);
  m2.write(45);
//segundo siatem
  m3.write(0);
  delay(500);
  m3.write(45);
  delay(50);
  m3.write(0);
  delay(500);
  m3.write(45);
  m4.write(17);
  delay(500);
  m4.write(45);
  m5.write(10);
  delay(500);
  m5.write(25);
  delay(100);
  m5.write(10);
  delay(500);
  m5.write(25);
  //primer compas
  m4.write(17);
  delay(500);
  m4.write(45);
  m3.write(0);
  delay(500);
  m3.write(45);
  m2.write(17);
  delay(500);
  m2.write(45);
```

```
  //segundo compas
  m1.write(0);
  delay(500);
  m1.write(45);
  delay(50);
  m1.write(0);
  delay(500);
  m1.write(45);
  m2.write(17);
  delay(500);
  m2.write(45);
  m3.write(0);
  delay(500);
  m3.write(45);
  m2.write(17);
  delay(500);
m2.write(45);
m1.write(0);
  delay(500);
  m1.write(45);
    delay(50);
 m1.write(0);
delay(500);
  m1.write(45);
//  rotate(10*tono,1);
}

void findZero()
{
  limit= true;
  while(limit)
  {
    rotate(50, .1);

    if(digitalRead(7)==0)
      limit=false;
  }
}

void rotate(int steps, float speed){
  //rotate a specific number of microsteps (8 microsteps per step) - (negitive for reverse
movement)
  //speed is any number from .01 -> 1 with 1 being fastest - Slower is stronger
  int dir = (steps > 0)? HIGH:LOW;
  steps = abs(steps);
```

```
  digitalWrite(DIR_PIN,dir);

  float usDelay = (1/speed) * 70;

  for(int i=0; i < steps; i++){
    digitalWrite(STEP_PIN, HIGH);
    delayMicroseconds(usDelay);

    digitalWrite(STEP_PIN, LOW);
    delayMicroseconds(usDelay);
  }
}
```