Politecnico di Milano, 2016-05-18
Robotics and Design 4th Edition
2Shy – Antonia Rekman, Bob (Bao) Shunlei,
Nima Maleki, Paolo Eugenio Galluzzi, Simone Deola

2Shy™ Robot

"The Final Report"

1 Design & Development of the robot

1.1 Decision on Music and style

The group decided to develop a robot that represents the Hip-Hop music style. At first, the decision was beat-boxing, but since it is difficult to apply a certain style and characteristics to beat-boxing, or even call it a defined music style, the Hip-Hop alternative was more appealing. In order to further specify which kind of Hip-Hop the robot should represent all group members spent some time watching Hip-Hop videos and listening to different kinds of Hip-Hop music. A mood-board was also created at this point in the process. The final decision fell on old-school Hip-Hop, since the group members liked this music the most and also found the characteristics of graffiti, boom-boxes, old cars and baggy clothes appealing.

1.2 Initial designing process

Many different designs were discussed within the group, and all group members came up with their own sketches on a first design. The winning sketches assembled the old-school Hip-Hop in details inspired from both a low-rider car and a boom-box. All group members agreed upon the boom-box as the main inspiration for the design and started to work with deciding which parts should be

included. The designer in the group made sketches and the work with the mechanical design started as soon as the movements of the robot had been decided.

Early on the group decided that the robot should have wheels as the main enabler for the movements. In the beginning the group also discussed a contracting movement supported by springs but left that idea since the wheels were a more convenient alternative, and the contracting movement was also feasible by using them instead.

1.3 Decision on movements and emotions

Since Hip-Hop music mainly focuses on heavy beats and does not include so many other elements, the group found it important that the movement of the robot should exactly correspond to the beats and the tempo with one big characteristic movement instead of a large set of many small movements. This is also inspired from the real Hip-Hop scene, where Hip-Hop performers are often seen to do just one kind of movement, typically with their hand up and down. A good example would be the street dance called the "Gangsta Two-Step".

The group had the idea that the robot, which has the shape of a boom-box, could move the two "speakers" back and forth to represent the movement of the shoulders on a person. Moreover, a real boom-box has a "door" for cassettes that opens and closes. Both of these movements also inspired the movements of the robot.

Moreover, two movements were added which do not happen to be originated from a boom-box, but instead from a human hip-hop dancer and DJ. Hip-hop dancers are known to move their baseball caps left and right while dancing. This is exactly translated to the cap that is visible on top of the robot and moves with the rhythm. The other movement is inspired by the act of playing with an old style disc on a turntable by a hip-hop DJ to create a sound effect, called "scratching". This movement is realized by inserting an equivalent disc which turns and responds to the mid frequencies. although the two latter movements do not happen as frequent as the movement of the wheels and the cassette door, which was of course on purpose.

After all, an LED stripe was designed to serve as the "eyes" of the robot and also an equalizer or VU-Meter. When the robot is idle, the robot "blinks" and looks around to find music, and when it hears the music it wants, it starts to show a light effect through the LED stripe. The LED-light was first thought to be added to the "speaker" in a circle but this solution was harder to achieve and also more expensive, moreover the lights in a row looked more like the old-school boom-box that the group had in mind from the initial moodboard.

1.4 Mechanical Design

Since the group aimed for a contracting movement enabled by wheels, the base of the robot had to be divided into 3 parts, one middle part and two, moving side parts. The front of the robot needed to have an opening for the "cassette door" that could move as well. The spinning "turntable disc" on the left side needed to be considered too. But, the group did not want to make a rectangular shape of the robot since this should have made the robot look too much like a dead and still object. In order to avoid the look of an immovable object, the designer in the group sketched the robot anew as a boom-box but with hexagonal shapes of the two side parts. This made the robot look more like an active object and gave it character. The decision came to produce a prototype of the "shell" for the robot in thin wood, since the material was easy to work with and the designer in the group had previous experience in working with it.

In terms of aesthetics, the wood was treated to look like a metallic material with steel-coloured spraypaint. The inspiration from low-rider cars was also carried out in a design of golden colour on the "shell" of the robot. On the right side is the "speaker" that is basically constructed to look like a speaker but has no connected sound-system or movement.

On the upper side of the LED stripe, there's a graffiti "tag", which reads "2SHY". This kind of tagging is used widely in hip-hop culture alongside with graffities themselves. The tag was printed on paper and then applied directly on wood using stencil technique.
The turntable disc shows a thin golden line and also a phrase reading "All eyes on me" which is the title of a phenomenal 1996 hit song by legendary rapper 2Pac. The rapper's name has an irony with the robot's name, as well as the phrase "All eyes on me" with the robot's partially shy character. This was done following professor Bonarini's advise on making the disc's movement more visible.

1.5 Problems and Solutions in the design

In all, the mechanical design of the robot is quite simple. Focus has been on producing a working robot and the group has made an effort to not complicate the design and rather start with simpler ideas and then develop them, like when the spinning cap was added. The goal has been to have a robot that works in 100% and therefore the design has been kept simple but effective.

The main challenge in the mechanical design was to find the solution for the attachment of the moving side parts to the middle part. This was solved by adding two vertical axis which allow the side parts to move back and forth. On the bottom of each side part is also a wheel with an independent servo-motor that allows the wheel to spin in its own direction and hence move the two sides separately from each other. But the two vertical axis, that are attached to the bottom board and the top board, also created a challenge for how to easily assemble and de-assemble the robot.

Another problem that needed to be solved was the opening/closing movement of the "cassette" door. This was solved by adding a spring so that the initial state of the door is to be open, and then is being pulled in by a string attached to the servo-motor.

2 Technical manual

2.1 Parts

1.1 BODY OUTSIDE

1.1.1 BOTTOM MIDDLE OUTSIDE
1.1.1.1 - 2 omniwheels + screws
1.1.1.2 - 4 bolts (medium)

1.1.2 BOTTOM LEFT OUTSIDE
1.1.2.1 - 4 screws (medium)
1.1.2.2 - Left wheel
1.1.2.3 - Plastic cross
1.1.2.4 - 2 screws (mini)
1.1.2.5 - Servo screw

1.1.3 BOTTOM RIGHT OUTSIDE
1.1.3.1 - 4 screws (medium)
1.1.3.2 - Left wheel
1.1.3.3 - Plastic cross
1.1.3.4 - 2 screws (mini)
1.1.3.5 - Servo screw

1.1.4 FRONT MIDDLE OUTSIDE
1.1.4.1 - 2 front middle folding pieces
1.1.4.2 - LED-window (plastic)
1.1.4.3 - 2 screws
1.1.4.4 - "Casssette door"
1.1.4.4.1 - Hinge + 4 screws
1.1.4.4.2 - 2 springs
1.1.4.4.3 - String + bar
1.1.4.4.4 - "Cassette"

1.1.5 LEFT FRONT OUTSIDE
1.1.5.1 - Disc + "reader"
1.1.5.2 - plastic cross
1.1.5.3 - 4 screws (medium)
1.1.5.4 - Servo screw
1.1.5.5 - Left front folding piece (paper)

1.1.6 RIGHT FRONT OUTSIDE
1.1.6.1 - "Speaker"
1.1.6.2 - 2 screws (medium)
1.1.6.3 - Right front folding piece (paper)

1.1.7 BACK MIDDLE OUTSIDE
1.1.7.1 - 2 back middle folding pieces (paper)
1.1.7.2 - Mic (sensor)
1.1.7.3 - Power-jack
1.1.7.4 - Power-switch
1.1.7.5 - Corner-piece (metal)
1.1.7.6 - screw + bolt (medium)

1.1.8 LEFT BACK OUTSIDE
1.1.8.1 - 2 screws (medium)
1.1.8.2 - Left back folding piece (paper)

1.1.9 RIGHT BACK OUTSIDE
1.1.9.1 - 2 screws (medium)
1.1.9.2 - Right back folding piece (paper)

1.1.10 TOP MIDDLE OUTSIDE
1.1.10.1 - Block (foam)
1.1.10.2 - Servo-motor
1.1.10.3 - "Head-cross" (foam)
1.1.10.3.1 - Button (sensor)
1.1.10.3.2 - Cross (plastic) + 2 screws (medium)

1.1.11 TOP LEFT OUTSIDE
1.1.12 TOP RIGHT OUTSIDE

1.2 BODY INSIDE

1.2.1 BOTTOM MIDDLE INSIDE
1.2.1.1 - 2 axis + bolts (medium)
1.2.1.2 - 2 corner-pieces + bolts
1.2.1.3 - 4 bolts (mini)

1.2.2 BOTTOM LEFT INSIDE
1.2.2.1 - Servo-motor + 2 bolts
1.2.2.2 - Block (foam)
1.2.2.3 - 2 corner-pieces + bolts
1.2.2.4 - 2 blocks (wood)

1.2.3 BOTTOM RIGHT INSIDE
1.2.3.1 - Servo-motor + 2 bolts
1.2.3.2 - Block (foam)
1.2.3.3 - 2 corner-pieces + bolts
1.2.3.4 - 2 blocks (wood)

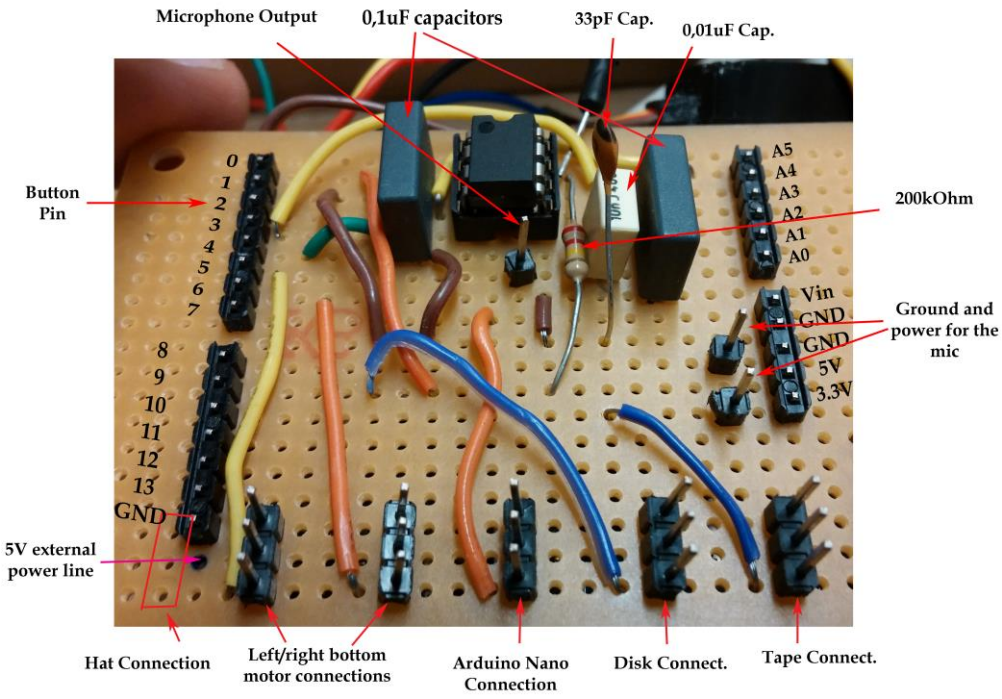
1.2.4 FRONT MIDDLE INSIDE
1.2.4.1 - LED-lights
1.2.4.2 - Back of "cassette" piece (wood)
1.2.4.3 - 4 bolts (medium)
1.2.4.4 - 2 screws (mini)

1.2.5 FRONT LEFT INSIDE
1.2.5.1 - Servo-motor (micro)
1.2.5.2 - Block (foam)

see next page for the electric pack...!

1.4 Electric pack

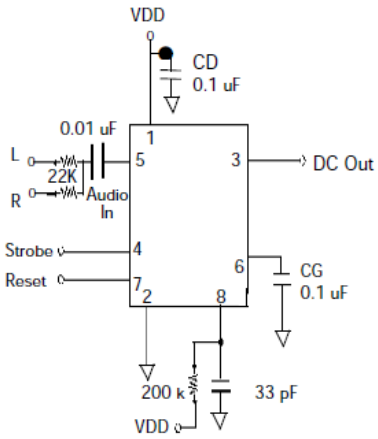
1.4.1 Arduino Uno
1.4.2 Arduino Nano
1.4.3 33 pF capacitor
1.4.4 2 x 0,1 uF capacitor
1.4.5 200 kΩ resistance
1.4.6 0,01uF capacitor
1.4.7 msgeq7- chip
1.4.8 Cables



2.2 Electric Circuit choices and description

The main circuit on the protoboard was designed to act like a shield, which makes the electric pack more compact and simpler to connect and handle.

Since the main purpose of the robot was to listen to sounds and analyse them, we choose a mic with gain-control (MAX4466) and a 7-band graphic equalizer (MSGEQ7). Making the Fourier-transform process directly in the Arduino Uno is a hard and time-consuming task, so that is why we choose the msgeq7-chip. The image of the circuit we implemented around the chip is seen to the right.[1]
The chip capture an input and make a filtering process (for example using a first low-pass band filter). The output is stored in one variable of the Arduino uno, and a strobe "impulse" switch the output of the MSGEQ7 to the next filter, so we can read the value from another band. It's a multiplexer so you cannot read all the values at the same time (we have only one output from the chip).



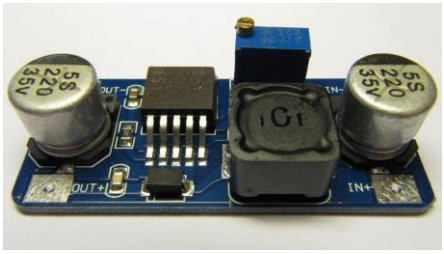
Picture 2.2 msgeq7-chip circuit

The outcome of the process is then used to manipulate the different servos according to the different frequencies. For example the base-part is moving according to frequencies in the lower band-width.

To power the system, a division is made between the power for the Arduino Uno and the power for the servos. This is essential since the servos are connected with the LED-lights that support a maximum of 5 V and the Arduino Uno supports a maximum of 18 V in the V-In pin. Also the Arduino haven't got enough power to manage all the motors.

[1] https://www.sparkfun.com/datasheets/Components/General/MSGEQ7.pdf

The input is divided using a tension regulator (LM 2577, see the the picture on the right).
The output from the tension regulator is sent to the servos, to the LED-lights and to the Arduino Nano using the bottom pins of the circuit show in the first picture; While the same input of the LM2577 goes to the Vin pin of the Arduino UNO, to power up all the robot's brain.



Notes:

Since the mic is sensitive it had to be put in the 3.3 V pin of the Arduino Uno, that is less noisy. Moreover, there was initially a problem with shaky movements from the servo-motors, which was solved by adding and using the Arduino Nano for controlling the LED-lights.

The final day of the work process the push-button was added at the top of the spinning "head" of the robot, controlled by pin 2, with a resistor in parallel (not illustrated in the first picture).

2.3 Programming Description

2.4 Code

```
#include <VarSpeedServo.h>

int strobe = 4; // strobe pins on digital 3
int res = 3; // reset pins on digital 2
int analogPin = 5;


int leftP=0;
int rightP=1;
int leftpin = 5;
int rightpin = 6;
VarSpeedServo left;
VarSpeedServo right;

int diskP=0;
int diskPin=9;
VarSpeedServo disk;

int hatP=0;
int hatPin=11;
VarSpeedServo hat;



int doorP=0;
int doorPin=10;
int rnd=0;
VarSpeedServo door;

//int frontpin=9;
int spectrumValue[7]; // store band values
in these arrays
int band;
float low = 0;
float mid = 0;


int r=0;
int g=0;
int b=0;

int light =7;
unsigned long chorTime=0;
unsigned long busyTime=0;
int interval = 700;
bool beat=false;
int average=0;

int moveSet=0;
int moveSetStep=0;
int level=0;
unsigned long busyDoorTime=0;
unsigned long busyDiskTime=0;
int intervalDoor=200;
int turntableInterval=50;

bool voice=false;
int lightState=1;
int pat=0;
int interruptPin=2;



//thresholds
int midTh=750;
int beatTh=750;



//initialization of all the things
void setup()
{
  Serial.begin(9600);
  pinMode(analogPin, INPUT);
  pinMode(res, OUTPUT); // reset
  pinMode(strobe, OUTPUT); // strobe
  digitalWrite(res, LOW); // reset low
  digitalWrite(strobe, HIGH); //pin 11 is
RESET on the shield
  left.attach(leftpin);
  right.attach(rightpin);
  pinMode(light, OUTPUT);
  left.write(90,255,true);
  right.write(90,255,true);

  door.attach(doorPin);
  door.write(10,255,true);
  disk.attach(diskPin);
  disk.write(10,255,true);
  hat.attach(hatPin);
  hat.write(10,255,true);


  pinMode(interruptPin, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(in-
terruptPin), increasePat, RISING);


}

void increasePat(){
  pat++;
}

void readMSGEQ7()
// Function to read 7 band equalizers
{
  digitalWrite(res, HIGH);
  digitalWrite(res, LOW);
  for (band = 0; band < 7; band++)
  {
    digitalWrite(strobe, LOW); // strobe pin
on the shield - kicks the IC up to the next
band
    delayMicroseconds(30); //
    spectrumValue[band] = analogRead(analog-
Pin); // store band reading

    digitalWrite(strobe, HIGH);
  }
}
```

```
//it returns 1 if the arduinos detect a beat, it works by detecting a peak in the average value of the low bands.
bool beatDetect(){
  low    =    (spectrumValue[2]    +    spectrumValue[3]) / 2;
  if (low > beatTh) {
   return true;
    }
    else{
      return false;
    }

}

//it returns 1 if the arduino detect the voice in the song, it works by detecting peaks in the average value the medium bands.
bool voiceDetect(){
  mid    =    (spectrumValue[4]    +    spectrumValue[5]) / 2;
    //Serial.println(low);
  if (mid > midTh ) {
   return true;
    }
    else{
      return false;
    }
}

//it detect the level of the sound, it's the average value of all the bands mapped with value from 0 to 3.
int levelDetect(){
  average=0;
  for(int i=0; i<7;i++){
      average=average+spectrumValue[i];
  }
  average= average/7;
  if(average <250){
    return 0;
  }
  if(average >=250 && average<= 350){
    return 1;
  }
    if(average >350 && average<= 900){
    return 2;
  }
    if(average >900){
    return 3;
  }
}

//return true if the robot is busy. to do that check that is passed a certain ammount of time (interval) since the last
//time i do something with the robot (busyTime).
bool busy(){
  return ( millis()< busyTime +interval);
}

//return true if the door is busy.
bool doorBusy(){
  return ( millis()< busyDoorTime +intervalDoor);
}

//return true if the disk (or the hat) is busy.
bool diskBusy(){
  return ( millis()< busyDiskTime +turntableInterval);
}
```

```
//all the action to move the wheels
void moveWheels(int level){


  moveSet=(random(100)%4) +1;
  moveSet=2;
  moveSetStep=0;
  switch(moveSet){
    case 1:
        right.write(150,255,false);
        left.write(150,255,false);
      break;
    case 2:
        right.write(150,155,false);
        left.write(150,155,false);
      break;
    case 3:
        right.write(150,255,false);
        left.write(150,255,false);
      break;
    case 4:
       right.write(150,50,false);
       left.write(150,50,false);
      break;
  }

  /*rnd=random(2);
  switch (level){
    case 0:
      //DONOTHING
      break;
    case 1:


      break;
    case 2:


      break;
    case 3:
      break;
    default:
      break;
  }*/
}

//all the movement to move the door
void moveDoor(int level){

  door.write(doorP*160+10, 255, false);
  doorP=1-doorP;

  /*switch (level){
    case 0:
      break;
    case 1:
      break;
    case 2:
      break;
    case 3:
      break;
    default:
      break;
  }  */
}




//all the movement to move the disk
void moveDisk(int level){

  disk.write(doorP*160+10, 255, false);
```

```
  diskP=1-diskP;
   /*switch (level){
    case 0:
      break;
    case 1:
      break;
    case 2:
      break;
    case 3:
      break;
    default:
      break;
  }*/
}

//all the movement of the hat.
void moveHat(int level){

  hat.write(hatP*160+10, 255, false);
  hatP=1-hatP;
}

//send a message to the NANO that make the
'beat' choreography in the LED
void lightUp(int level){
  digitalWrite(light, HIGH);
  delay(10);
  digitalWrite(light, LOW);
  lightState=1-lightState;
  if(lightState==0){
    digitalWrite(light, LOW);
  }else{
    digitalWrite(light, HIGH);
  }

}

void moveSetChecker(){
  //'return' action for the first choreogra-
phy.
  if(moveSet==1){
    if(left.read()>=150){
        right.write(90,25,false);
        left.write(90,25,false);
        moveSet=0;
      }

  //'return' action for the second choreog-
raphy.
  }if(moveSet==2){
    if(left.read()>=150 && moveSetStep==0){
      moveSetStep++;
      chorTime=millis();
    }

    if(left.read()>=150  &&  millis()>chor-
Time+100 && moveSetStep==1){
      right.write(130,255,false);
      left.write(130,255,false);

    }
    if(left.read()==130 && moveSetStep==1){
      chorTime=millis();
      moveSetStep++;
    }
    if(left.read()==130  &&  millis()>chor-
Time+100 && moveSetStep==2){
      right.write(110,255,false);
      left.write(110,255,false);

    }
   if(left.read()==110 && moveSetStep==2){
      chorTime=millis();
      moveSetStep++;
    }
    if(left.read()==110&&    millis()>chor-
Time+100 && moveSetStep==3){
      right.write(90,255,false);
      left.write(90,255,false);
      moveSet=0;
      moveSetStep=0;
    }
  }



//'return' action for the third choreogra-
phy.
  if(moveSet==3){
    if(left.read()>=150){
        right.write(90,255,false);
        left.write(90,255,false);
        moveSet=0;
      }
  }

  //'return' action for the fourth choreog-
raphy.
  if(moveSet==4){
    if(left.read()>=150){
        right.write(90,50,false);
        left.write(90,50,false);
        moveSet=0;
      }
  }
}


void loop()
{
  if(pat<20){
      //REGULAR PART (NOT THE WINNING ONE)
      //read the value from the microphone
    readMSGEQ7();
    //check if there is a beat and save the
result in the beat variable.
    beat=beatDetect();
    //calculate the volume of the sound and
save it in level
    level= levelDetect();
    //check if there is the voice in the song
and save the result in the voice variable.
    voice=voiceDetect();

    //movement for the door
    if(voice && !doorBusy()){
      moveDoor(level);
      busyDoorTime=millis();
    }

    //movement for the disk and/or the hat
    if(voice && !diskBusy()){
      moveDisk(level);
      busyDiskTime=millis();
      moveHat(level);

    }

    //this part is executed when the beat
comes and only if i'm not busy
    if( !busy() && beat && left.read()==90
){

      //save that i have do something right
now
      busyTime=millis();
```

```
      //make the all movements
      moveWheels(level);

      moveDisk(level);


      lightUp(level);

    }


//this part finish the choreograpy started
when i set it in the function moveWheels,
when it have to be finished
    moveSetChecker();
  }else{
    //WINNING PART
    lightUp(level);
    delay(50);
    lightUp(level);
    delay(50);
    lightUp(level);
    delay(50);
    lightUp(level);
    delay(50);

    //return to the normal state.
    pat=0;
  }

}
```

3 Budget

Where	What	Price (€)
Zara Piazza Duomo	Small kids hat	6,99
http://www.robot-italy.com	Electret Microphone Amplifier - MAX4466	7,93
http://www.robot-italy.com	msgeq7- chip	5,86
www.robot-italy.com	3 x Hitech Microservo	8,93
Protoshop	Laser-cut	34,9
ProtoShop	Glue	4
Futura Elettronica	2 x Uni-cast wheel	6
Futura Elettronica	2 x Servo - 207 33 pF capacitor 2 x 0,1 uF capacitor 0,01uF capacitor 200 kΩ resistance	16
www.amazon.com	Neo pixel LED	8,79
Protoshop	2nd cut + paint + papers	42
www.ebay.com	tension regulator	5,5
www.ebay.com	PROTOBOARD	4
www.amazon.com	Arduino Nano Compatible	2,7
		Total € 182,45

4 Users Guide

See attached PDF.