```
// A Listening BEEST on Table: with a sound detector
// Copyright (C) 2017 ArduinoDeXXX All Rights Reserved.
//
// Materials:
// "Arduino NANO",
// A motor driver IC "L298N"
// A cheap sound detection module with 3 pins (for US$1)
// http://www.ebay.com/itm/High-Sensitive-Sound-Detection-Module-Voice-Control-Switch-Microphone-Sensor-Ard-/201052209891
//
// Information:
// 1S LiPo battery (3.7V) can be used for 2 DC motors. Connect its positive to pin4 on L298N.
// 6P battery (9V) can be used for Arduino NANO. Connect its positive to VIN-pin on NANO.
// Connect Vcc-pin on the sound detection module to 5V-pin on NANO.
// Connect Out-pin on the sound detection module to A0-pin on NANO.
//
// How to command:
// See the video below (0:22-).
// https://www.youtube.com/watch?v=Zvt43uCgI4U
//
// View the sites bellow to see more detail.
// http://www.instructables.com/id/Training-Theo-Jansens-Mini-BEEST/
// http://www.instructables.com/id/Training-Theo-Jansens-Mini-BEEST-JPN/


int output;
const int getSignal = 700;
unsigned long recNoiseIn = 0;
unsigned long recTime = 0;
unsigned long recTempTime = 0;
const unsigned long t1L = 200;
const unsigned long t1U = 420;
const unsigned long t2L = 570;
const unsigned long t2U = 790;
int state = 0;
const unsigned long waiting = 150;

void setup() {
  pinMode( 9, OUTPUT );
  pinMode( 6, OUTPUT );
  pinMode( 7, OUTPUT );
  pinMode( 8, OUTPUT );
  pinMode( 5, OUTPUT ); //PWM1
  pinMode( 3, OUTPUT ); //PWM2
  digitalWrite( 9, LOW );
  digitalWrite( 6, LOW );
  digitalWrite( 7, LOW );
  digitalWrite( 8, LOW );
  digitalWrite( 5, HIGH );
  digitalWrite( 3, HIGH );

/* If you want to see beating rhythm before commanding, activate the 11 lines below.
  pinMode( 13, OUTPUT );
  digitalWrite( 13, HIGH );
  delay( 1000 );
  digitalWrite( 13, LOW );
  delay( 1000 );
  for ( int i=0 ; i<15 ; i++ ) {
    digitalWrite( 13, HIGH );
    delay( 50 );
    digitalWrite( 13, LOW );
    delay( 260 );
  }
*/
  delay( 1000 );
}


void loop() {
  output = analogRead(A0);
  if ( output > getSignal ) {
    if ( millis() - recNoiseIn < 10 ) {
```

```
      check();
    }
    recNoiseIn = millis();
  }

  if ( millis() - recTime > t2U ) {
    if      ( state == 1  ) { stopping(); }
    else if ( state == 2  ) { forward(); }
    else if ( state == 11 ) { right(); }
  }
  delay(1);
}


void check() {
  recTempTime = millis();
  if ( recTempTime - recTime > t1L ) {
    tempStop();
    if ( recTempTime - recTime < t1U ) {
      state++;
      recTime = recTempTime;
      if ( state > 2 ) { back(); }
    } else {
      if ( recTempTime - recTime < t2U ) {
        state = state + 10;
        recTime = recTempTime;
        if ( state > 20 ) { left(); }
      } else {
        state = 1;
        recTime = recTempTime;
      }
    }
  }
}

void forward() {
  digitalWrite( 8, HIGH );
  digitalWrite( 9, LOW );
  digitalWrite( 6, LOW );
  digitalWrite( 7, HIGH );
  recTime = 0;
  recTempTime = 0;
  state = 0;
}
void back() {
  digitalWrite( 8, LOW );
  digitalWrite( 9, HIGH );
  digitalWrite( 6, HIGH );
  digitalWrite( 7, LOW );
  recTime = 0;
  recTempTime = 0;
  state = 0;
  delay( waiting );
}
void tempStop() {
  digitalWrite( 8, LOW );
  digitalWrite( 9, LOW );
  digitalWrite( 6, LOW );
  digitalWrite( 7, LOW );
}
void stopping() {
  digitalWrite( 8, LOW );
  digitalWrite( 9, LOW );
  digitalWrite( 6, LOW );
  digitalWrite( 7, LOW );
  recTime = 0;
  recTempTime = 0;
  state = 0;
}
void right() {
  digitalWrite( 8, LOW );
```

```
  digitalWrite( 9, HIGH );
  digitalWrite( 6, LOW );
  digitalWrite( 7, HIGH );
  recTime = 0;
  recTempTime = 0;
  state = 0;
}
void left() {
  digitalWrite( 8, HIGH );
  digitalWrite( 9, LOW );
  digitalWrite( 6, HIGH );
  digitalWrite( 7, LOW );
  recTime = 0;
  recTempTime = 0;
  state = 0;
  delay( waiting );
}
```