Lesson 8- Robot action decomposition

1、Overview

In the sixth lesson we have successfully assembled the robot. Here is the code for driving the robot. A sample code is also provided for testing. Please install libraries files to provide library files before using.

2、Code Decomposition

```
#include <Servo.h>         //servo library, for driving the servo motor rotation function
#include <Oscillator.h>    //robot library, contains various robot action functions
#include <EEPROM.h>        // Chip EEPROM read and write library, EEPROM read and write operation function

#define N_SERVOS 4         //Define N_SERVOS as the number 4
//-- First step: Configure the pins where the servos are attached
/*
          ---------------
         |     0   0     |
         |---------------|
YR 6==> |               | <== YL 11
          ---------------
             ||     ||
             ||     ||
RR 7==>    -----   ------  <== RL 10
          |-----   ------|
*/
#define EEPROM_TRIM false      //Define EEPROM_TRIM as no
// Activate to take callibration data from internal memory
#define TRIM_RR 0
#define TRIM_RL 0
#define TRIM_YR 0
#define TRIM_YL 0
//OTTO.setTrims(-7,-4,-4,7);

#define PIN_RR 7       //Define PIN_RR as the number 7
#define PIN_RL 10      //Define PIN_RL as the number 10
#define PIN_YR 6       //Define PIN_YR as the number 6
#define PIN_YL 11      //Define PIN_YL as the number 11

#define INTERVALTIME 10.0     //Define INTERVALTIME as the number 10.0

Oscillator servo[N_SERVOS];   //Define the Oscillator class array servo[4]

void goingUp(int tempo);      //Declaration function goingUp ()
void drunk (int tempo);       //Declaration function drunk ()
```

```
void noGravity(int tempo);
void kickLeft(int tempo);
void kickRight(int tempo);
void run(int steps, int T=500);
void walk(int steps, int T=1000);
void backyard(int steps, int T=3000);
void backyardSlow(int steps, int T=5000);
void turnLeft(int steps, int T=3000);
void turnRight(int steps, int T=3000);
void moonWalkLeft(int steps, int T=1000);
void moonWalkRight(int steps, int T=1000);
void crusaito(int steps, int T=1000);
void swing(int steps, int T=1000);
void upDown(int steps, int T=1000);
void flapping(int steps, int T=1000);

void setup()
{
  Serial.begin(19200);         //Set the serial port baud rate to 19200

  pinMode(12,OUTPUT);          //Set D12 pin to output mode
  pinMode(13,OUTPUT);          //Set D13 pin to output mode
  digitalWrite(12,HIGH);       //Set D 12 pin output high level.
  digitalWrite(13,HIGH);       //Set D13 pin output high level

  servo[0].attach(PIN_RR);     //Defining D7 to control the servo of the robot's right foot
  servo[1].attach(PIN_RL);     //Defining D10 to control the servo of the robot's left foot
  servo[2].attach(PIN_YR);     //Defining D6 to control the servo of the robot's right foot
  servo[3].attach(PIN_YL);     //Defining D11 to control the servo of the robot's left foot

  //EEPROM.write(0,TRIM_RR);
  //EEPROM.write(1,TRIM_RL);
  //EEPROM.write(2,TRIM_YR);
  //EEPROM.write(3,TRIM_YL);

  int trim;                    //Define the shaping character trim

  if(EEPROM_TRIM){             //Judgment statement, used to determine whether to read the data of the chip
EEPROM, here is no
    for(int x=0;x<4;x++){
      trim=EEPROM.read(x);
      if(trim>128)trim=trim-256;
      Serial.print("TRIM ");
      Serial.print(x);
      Serial.print(" en ");
      Serial.println(trim);
      servo[x].SetTrim(trim);
    }
  }
  else{
```

```
    servo[0].SetTrim(TRIM_RR);    //Enable D7 to control the servo of the right foot
    servo[1].SetTrim(TRIM_RL);    //Enable D10 to control the servo of the left foot
    servo[2].SetTrim(TRIM_YR);    //Enable D6 to control the servo of the right foot
    servo[3].SetTrim(TRIM_YL);    //Enable D11 to control the servo of the left foot
  }

  for(int i=0;i<4;i++) servo[i].SetPosition(90);  //Initialize all servos of robot to 90 degrees
}

// TEMPO: 121 BPM
int t=495;       //Define t as the number 495
double pause=0;  //Define the double precision character pause as the number 0

void loop()     //The main loop function, arduino code must have this function, some function implementation functions can be placed inside
{
 // if(Serial.available()){
  //  char init = Serial.read();
   // if (init=='X'){
   //   delay(4000); //3000 - 4500

dance();   //Robot dance function, robot has been loop through this dance function


//for(int i=0;i<4;i++) servo[i].SetPosition(90);


          for(int i=0;i<4;i++) servo[i].SetPosition(90);  //Initialize four servos of robot to 90 degrees
   // }
 // }
}

void dance(){               //Subfunctions of robot Dance
  primera_parte();          //Call the function, the left and right feet of the robot are lifted inward alternately
  segunda_parte();          //Call the function, the left and right feet of the robot are lifted inward at the same time
  moonWalkLeft(4,t*2);      //Call the function, the robot moves to the left in the space dance step, the number 4 is the number of moving steps, t is the time
  moonWalkRight(4,t*2);     //Call the function, the robot moves to the right in the space dance step, the number 4 is the number of moving steps, t is the time
  moonWalkLeft(4,t*2);
  moonWalkRight(4,t*2);
  primera_parte();
  crusaito(1,t*8);          //Call the function, the robot moves to the right in the space dance step slowly
  crusaito(1,t*7);

  for (int i=0; i<16; i++){
    flapping(1,t/4);                  //Call the function,lift the feet of the robot inward quickly
    delay(3*t/4);
```

```
  }

  moonWalkRight(4,t*2);
  moonWalkLeft(4,t*2);
  moonWalkRight(4,t*2);
  moonWalkLeft(4,t*2);

  drunk(t*4);                          //Call the function,robot swings left and right
  drunk(t*4);
  drunk(t*4);
  drunk(t*4);
  kickLeft(t);                        //Call the function,robot raises the right foot
  kickRight(t);                       //Call the function,robot raises the right foot
  drunk(t*8);
  drunk(t*4);
  drunk(t/2);
  delay(t*4);

  drunk(t/2);

  delay(t*4);
  walk(2,t*2);                        //Call the function,the robot moves forward
  backyard(2,t*2);                    //Call the function,the robot moves backward
  goingUp(t*2);                       //Call the function,the robot jumps
  goingUp(t*1);
  noGravity(t*2);                     //Call the function,the robot body tilt
  crusaito(1,t*2);
  crusaito(1,t*8);
  crusaito(1,t*2);
  crusaito(1,t*8);
  crusaito(1,t*2);
  crusaito(1,t*3);

  delay(t);
  primera_parte();
    for (int i=0; i<32; i++){
    flapping(1,t/2);
    delay(t/2);
  }

  for(int i=0;i<4;i++) servo[i].SetPosition(90);
}




////////////////////////////////////////////////////////////////////////////////
/////////////////////////////////FUNCIONES DE CONTROL//////////////////////////////
////////////////////////////////////////////////////////////////////////////////
//Servo motor initialization function
```

```
void oscillate(int A[N_SERVOS], int O[N_SERVOS], int T, double phase_diff[N_SERVOS]){
  for (int i=0; i<4; i++) {
    servo[i].SetO(O[i]);
    servo[i].SetA(A[i]);
    servo[i].SetT(T);
    servo[i].SetPh(phase_diff[i]);
  }
  double ref=millis();
   for (double x=ref; x<T+ref; x=millis()){
     for (int i=0; i<4; i++){
        servo[i].refresh();
     }
  }
}

unsigned long final_time;
unsigned long interval_time;
int oneTime;
int iteration;
float increment[N_SERVOS];
int oldPosition[]={90,90,90,90};

void moveNServos(int time, int  newPosition[]){        //Servo rotation function
  for(int i=0;i<N_SERVOS;i++)   increment[i] = ((newPosition[i])-oldPosition[i])/(time/INTERVALTIME);

  final_time =  millis() + time;

  iteration = 1;
  while(millis() < final_time){ //Javi del futuro cambia esto
      interval_time = millis()+INTERVALTIME;

      oneTime=0;
      while(millis()<interval_time){
          if(oneTime<1){
              for(int i=0;i<N_SERVOS;i++){
                  servo[i].SetPosition(oldPosition[i] + (iteration * increment[i]));
              }
              iteration++;
              oneTime++;
          }
      }
  }

  for(int i=0;i<N_SERVOS;i++){
    oldPosition[i] = newPosition[i];
  }
}


//////////////////////////////////////////////////////////////////////////////
```

```
////////////////////////////////PASOS DE BAILE////////////////////////////////
////////////////////////////////////////////////////////////////////////////////

void goingUp(int tempo){          //向上跳函数

      pause=millis();
      for(int i=0;i<4;i++) servo[i].SetPosition(90);
      delay(tempo);
      servo[0].SetPosition(80);
      servo[1].SetPosition(100);
      delay(tempo);
      servo[0].SetPosition(70);
      servo[1].SetPosition(110);
      delay(tempo);
      servo[0].SetPosition(60);
      servo[1].SetPosition(120);
      delay(tempo);
      servo[0].SetPosition(50);
      servo[1].SetPosition(130);
      delay(tempo);
      servo[0].SetPosition(40);
      servo[1].SetPosition(140);
      delay(tempo);
      servo[0].SetPosition(30);
      servo[1].SetPosition(150);
      delay(tempo);
      //servo[0].SetPosition(20);
      //servo[1].SetPosition(160);
      //delay(tempo);

      while(millis()<pause+8*t);

}

void primera_parte(){                     //The function of the robot raising the left and right feet
alternately

  int move1[4] = {60,120,90,90};
  int move2[4] = {90,90,90,90};
  int move3[4] = {40,140,90,90};

  for(int x=0; x<3; x++){
    for(int i=0; i<3; i++){
      lateral_fuerte(1,t/2);
      lateral_fuerte(0,t/4);
      lateral_fuerte(1,t/4);
      delay(t);
    }

    pause=millis();
```

```
    for(int i=0;i<4;i++) servo[i].SetPosition(90);
    moveNServos(t*0.4,move1);
    moveNServos(t*0.4,move2);
    while(millis()<(pause+t*2));
  }

  for(int i=0; i<2; i++){
    lateral_fuerte(1,t/2);
    lateral_fuerte(0,t/4);
    lateral_fuerte(1,t/4);
    delay(t);
  }

  pause=millis();
  for(int i=0;i<4;i++) servo[i].SetPosition(90);
  crusaito(1,t*1.4);
  moveNServos(t*1,move3);
  for(int i=0;i<4;i++) servo[i].SetPosition(90);
  while(millis()<(pause+t*4));
}

void segunda_parte(){                          //The function of the robot lifting the left and right feet
at the same time

  int move1[4] = {90,90,80,100};
  int move2[4] = {90,90,100,80};
  int move3[4] = {90,90,80,100};
  int move4[4] = {90,90,100,80};

  int move5[4] = {40,140,80,100};
  int move6[4] = {40,140,100,80};
  int move7[4] = {90,90,80,100};
  int move8[4] = {90,90,100,80};

  int move9[4] = {40,140,80,100};
  int move10[4] = {40,140,100,80};
  int move11[4] = {90,90,80,100};
  int move12[4] = {90,90,100,80};

  for(int x=0; x<7; x++){
    for(int i=0; i<3; i++){
      pause=millis();
      moveNServos(t*0.15,move1);
      moveNServos(t*0.15,move2);
      moveNServos(t*0.15,move3);
      moveNServos(t*0.15,move4);
      while(millis()<(pause+t));
    }
    pause=millis();
    moveNServos(t*0.15,move5);
```

```
    moveNServos(t*0.15,move6);
    moveNServos(t*0.15,move7);
    moveNServos(t*0.15,move8);
    while(millis()<(pause+t));
  }

  for(int i=0; i<3; i++){
    pause=millis();
    moveNServos(t*0.15,move9);
    moveNServos(t*0.15,move10);
    moveNServos(t*0.15,move11);
    moveNServos(t*0.15,move12);
    while(millis()<(pause+t));
  }
}

void lateral_fuerte(boolean side, int tempo){

  for(int i=0;i<4;i++) servo[i].SetPosition(90);
  if (side) servo[0].SetPosition(40);
  else servo[1].SetPosition(140);
  delay(tempo/2);
  servo[0].SetPosition(90);
  servo[1].SetPosition(90);
  delay(tempo/2);

}

void drunk (int tempo){                    //The function of the robot swinging left and right

  pause=millis();

  int move1[] = {60,70,90,90};
  int move2[] = {110,120,90,90};
  int move3[] = {60,70,90,90};
  int move4[] = {110,120,90,90};

  moveNServos(tempo*0.235,move1);
  moveNServos(tempo*0.235,move2);
  moveNServos(tempo*0.235,move3);
  moveNServos(tempo*0.235,move4);
  while(millis()<(pause+tempo));

}


void noGravity(int tempo){                      //The function of the robot body tilt

  int move1[4] = {110,130,90,90};
  int move2[4] = {130,130,90,90};
```

```
  int move3[4] = {110,130,90,90};
  int move4[4] = {90,90,90,90};


  for(int i=0;i<4;i++) servo[i].SetPosition(90);
  for(int i=0;i<N_SERVOS;i++) oldPosition[i]=90;
  moveNServos(tempo*2,move1);
  moveNServos(tempo*2,move2);
  delay(tempo*2);
  moveNServos(tempo*2,move3);
  moveNServos(tempo*2,move4);



}

void kickLeft(int tempo){                              //The function of lifting the left leg of the robot
  for(int i=0;i<4;i++) servo[i].SetPosition(90);
  delay(tempo);
  servo[0].SetPosition(50); //pie derecho
  servo[1].SetPosition(70); //pie izquiero
  delay(tempo);
  servo[0].SetPosition(80); //pie derecho
  servo[1].SetPosition(70); //pie izquiero
  delay(tempo/4);
  servo[0].SetPosition(30); //pie derecho
  servo[1].SetPosition(70); //pie izquiero
  delay(tempo/4);
  servo[0].SetPosition(80); //pie derecho
  servo[1].SetPosition(70); //pie izquiero
  delay(tempo/4);
  servo[0].SetPosition(30); //pie derecho
  servo[1].SetPosition(70); //pie izquiero
  delay(tempo/4);
  servo[0].SetPosition(80); //pie derecho
  servo[1].SetPosition(70); //pie izquiero
  delay(tempo);
}

void kickRight(int tempo){                              //The function of lifting the robot's right leg
for(int i=0;i<4;i++) servo[i].SetPosition(90);
  delay(tempo);
  servo[0].SetPosition(110); //pie derecho
  servo[1].SetPosition(130); //pie izquiero
  delay(tempo);
  servo[0].SetPosition(110); //pie derecho
  servo[1].SetPosition(100); //pie izquiero
  delay(tempo/4);
  servo[0].SetPosition(110); //pie derecho
  servo[1].SetPosition(150); //pie izquiero
```

```
  delay(tempo/4);
  servo[0].SetPosition(110); //pie derecho
  servo[1].SetPosition(80); //pie izquiero
  delay(tempo/4);
  servo[0].SetPosition(110); //pie derecho
  servo[1].SetPosition(150); //pie izquiero
  delay(tempo/4);
  servo[0].SetPosition(110); //pie derecho
  servo[1].SetPosition(100); //pie izquiero
  delay(tempo);
}

void walk(int steps, int T){                        //The function of the robot moving forward
    int A[4]= {15, 15, 30, 30};
    int O[4] = {0, 0, 0, 0};
    double phase_diff[4] = {DEG2RAD(0), DEG2RAD(0), DEG2RAD(90), DEG2RAD(90)};

    for(int i=0;i<steps;i++)oscillate(A,O, T, phase_diff);
}

void run(int steps, int T){                       //The function that the robot runs forward
    int A[4]= {10, 10, 10, 10};
    int O[4] = {0, 0, 0, 0};
    double phase_diff[4] = {DEG2RAD(0), DEG2RAD(0), DEG2RAD(90), DEG2RAD(90)};

    for(int i=0;i<steps;i++)oscillate(A,O, T, phase_diff);
}

void backyard(int steps, int T){                  //The function that the robot backward
    int A[4]= {15, 15, 30, 30};
    int O[4] = {0, 0, 0, 0};
    double phase_diff[4] = {DEG2RAD(0), DEG2RAD(0), DEG2RAD(-90), DEG2RAD(-90)};

    for(int i=0;i<steps;i++)oscillate(A,O, T, phase_diff);
}

void backyardSlow(int steps, int T){              //The function that the robot backward slowly
    int A[4]= {15, 15, 30, 30};
    int O[4] = {0, 0, 0, 0};
    double phase_diff[4] = {DEG2RAD(0), DEG2RAD(0), DEG2RAD(-90), DEG2RAD(-90)};

    for(int i=0;i<steps;i++)oscillate(A,O, T, phase_diff);
}

void turnLeft(int steps, int T){                  //The function that the robot turns left
    int A[4]= {20, 20, 10, 30};
    int O[4] = {0, 0, 0, 0};
    double phase_diff[4] = {DEG2RAD(0), DEG2RAD(0), DEG2RAD(90), DEG2RAD(90)};

    for(int i=0;i<steps;i++)oscillate(A,O, T, phase_diff);
```

```
}

void turnRight(int steps, int T){              //The function that the robot turns right
    int A[4]= {20, 20, 30, 10};
    int O[4] = {0, 0, 0, 0};
    double phase_diff[4] = {DEG2RAD(0), DEG2RAD(0), DEG2RAD(90), DEG2RAD(90)};

    for(int i=0;i<steps;i++)oscillate(A,O, T, phase_diff);
}

void moonWalkRight(int steps, int T){         //The function that the robot turns right with the moonwalk
    int A[4]= {25, 25, 0, 0};
    int O[4] = {-15 ,15, 0, 0};
    double phase_diff[4] = {DEG2RAD(0), DEG2RAD(180 + 120), DEG2RAD(90), DEG2RAD(90)};

    for(int i=0;i<steps;i++)oscillate(A,O, T, phase_diff);
}

void moonWalkLeft(int steps, int T){            //The function that the robot turns left with the moonwalk
    int A[4]= {25, 25, 0, 0};
    int O[4] = {-15, 15, 0, 0};
    double phase_diff[4] = {DEG2RAD(0), DEG2RAD(180 - 120), DEG2RAD(90), DEG2RAD(90)};

    for(int i=0;i<steps;i++)oscillate(A,O, T, phase_diff);
}

void crusaito(int steps, int T){                //The function that the robot turns right slowly with the
moonwalk
    int A[4]= {25, 25, 30, 30};
    int O[4] = {- 15, 15, 0, 0};
    double phase_diff[4] = {DEG2RAD(0), DEG2RAD(180 + 120), DEG2RAD(90), DEG2RAD(90)};

    for(int i=0;i<steps;i++)oscillate(A,O, T, phase_diff);
}

void swing(int steps, int T){                   //The function of robot march on the spot
    int A[4]= {25, 25, 0, 0};
    int O[4] = {-15, 15, 0, 0};
    double phase_diff[4] = {DEG2RAD(0), DEG2RAD(0), DEG2RAD(90), DEG2RAD(90)};

    for(int i=0;i<steps;i++)oscillate(A,O, T, phase_diff);
}

void upDown(int steps, int T){              //The function of the robot up and down
    int A[4]= {25, 25, 0, 0};
    int O[4] = {-15, 15, 0, 0};
    double phase_diff[4] = {DEG2RAD(180), DEG2RAD(0), DEG2RAD(270), DEG2RAD(270)};

    for(int i=0;i<steps;i++)oscillate(A,O, T, phase_diff);
}
```

```
void flapping(int steps, int T){            //The function of robot swing
    int A[4]= {15, 15, 8, 8};
    int O[4] = {-A[0], A[1], 0, 0};
    double phase_diff[4] = {DEG2RAD(0), DEG2RAD(180), DEG2RAD(90), DEG2RAD(-90)};

    for(int i=0;i<steps;i++)oscillate(A,O, T, phase_diff);
}

void test(int steps, int T){               //test function
    int A[4]= {15, 15, 8, 8};
    int O[4] = {-A[0] + 10, A[1] - 10, 0, 0};
    double phase_diff[4] = {DEG2RAD(0), DEG2RAD(180), DEG2RAD(90), DEG2RAD(-90)};

    for(int i=0;i<steps;i++)oscillate(A,O, T, phase_diff);
}
```