# Lesson 2 Make the Penguin Bot Move

## CONTENT



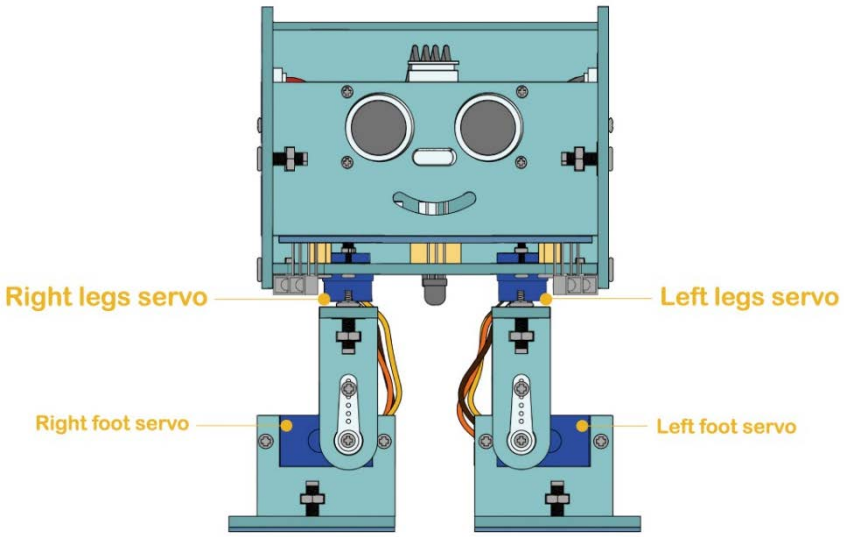
## Ⅰ. Brief Introduction

In this chapter you will learn the movement principle of Penguin Bot and how to program it. You will have a further understanding of the Penguin Bot after you finished this chapter.

## Ⅱ.Movement principle of Penguin Bot

### ***Walking Forwards***

Penguin Bot will need 4 beats to make one step forward and then recycle the 4 beats again to continue walking forwards.

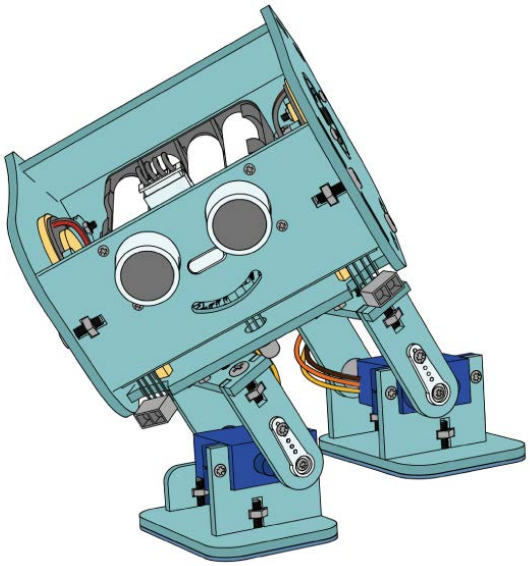



*Tips: If you have any questions or run into any problems during assembling and testing Penguin Bot please feel free to contact us at service@elegoo.com or euservice@elegoo.com (Europe customers).*

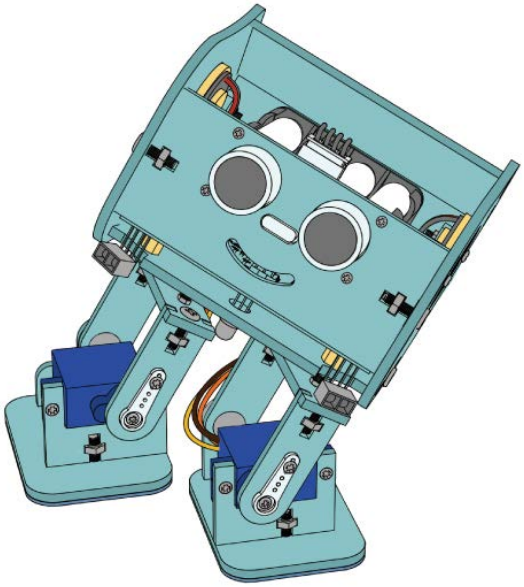
**Beats Diagram 1**

| Servo / Beats | Right Foot | Left Foot | Right Leg | Left Leg |
|---|---|---|---|---|
| 1st Beat | 120° | 120° | 60° | 60° |
| 2nd Best | 60° | 60° | 60° | 60° |
| 3rd Beat | 60° | 60° | 120° | 120° |
| 4th Beat | 120° | 120° | 120° | 120° |

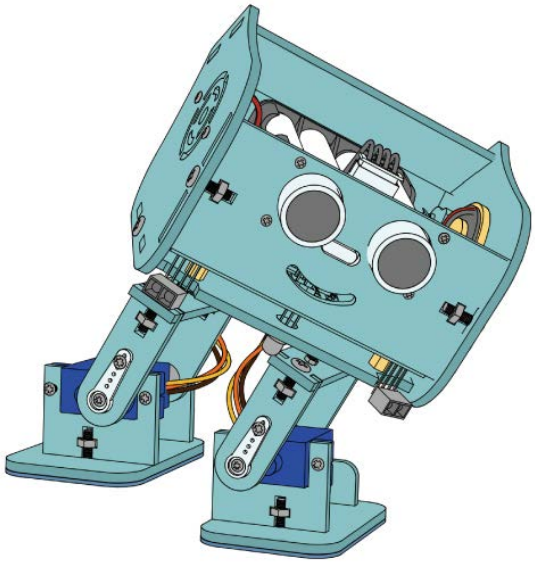
1st Beat: Servos of the left foot and right foot will rotate to 120° while servos of the left leg and right leg will rotate to 60°



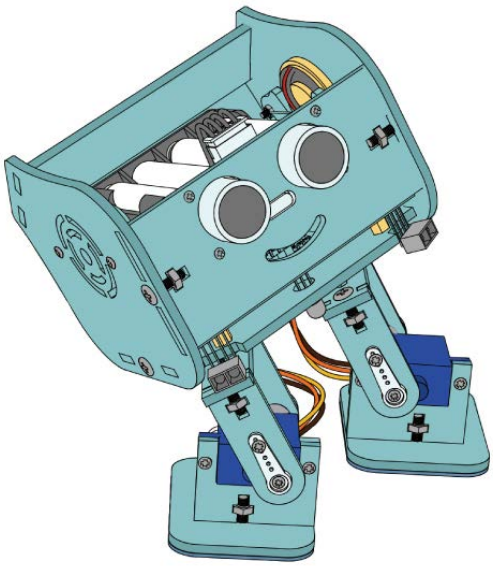
2nd Beat: Servos of the left foot and right foot will rotate to 60° while servos of the left leg and right leg will rotate to 60°



3rd Beat: Servos of the left foot and right foot will rotate to 60°while servos of the left leg and right leg will rotate to 120°



4th Beat: Servos of the left foot and right foot will rotate to 120°while servos of the left leg and right leg will rotate to 120°



According to the Beats Diagram, servos rotates by 30° at the position of 90°, then we used Oscillator libraries to control the beats of servo, which is a very easy and effective.

## ***Walking Backwards***

Execute the Beats Diagram 1 in a reversed order then you can make Penguin Bot walk backwards.

| Servo / Beats | Right Foot | Left Foot | Right Leg | Left Leg |
|---|---|---|---|---|
| **4th Beat** | 120° | 120° | 120° | 120° |
| **3rd Beat** | 60° | 60° | 120° | 120° |
| **2nd Best** | 60° | 60° | 60° | 60° |
| **1st Beat** | 120° | 120° | 60° | 60° |

## ***Turn Left***

Modify the left leg and right leg from 60° to 80° in the 1st and 2nd beat from Beats Diagram 1

| Servo / Beats | Right Foot | Left Foot | Right Leg | Left Leg |
|---|---|---|---|---|
| 1st Beat | 120° | 120° | 80° | 80° |
| 2nd Best | 60° | 60° | 80° | 80° |
| 3rd Beat | 60° | 60° | 120° | 120° |
| 4th Beat | 120° | 120° | 120° | 120° |

## ***Turn Right***

Modify the left leg and right leg from 120° to 100° in the 3rd and 4th beat from Beats Diagram 1

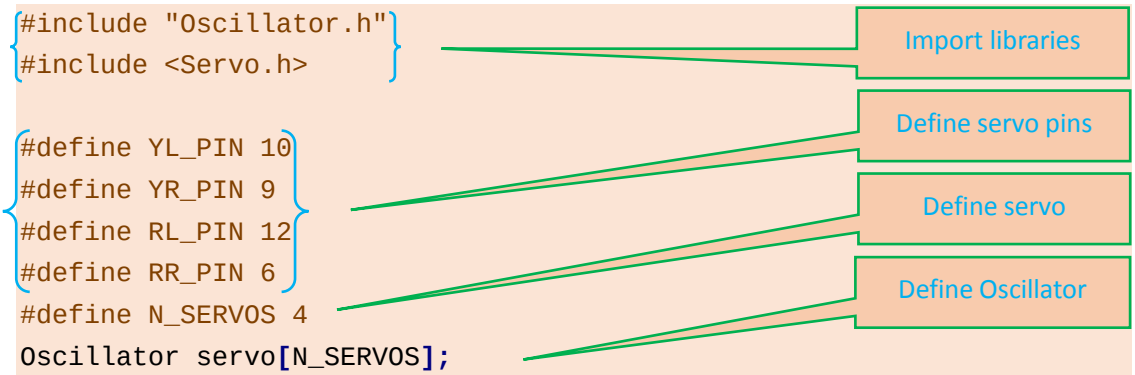
| Servo / Beats | Right Foot | Left Foot | Right Leg | Left Leg |
|---|---|---|---|---|
| 1st Beat | 120° | 120° | 60° | 60° |
| 2nd Best | 60° | 60° | 60° | 60° |
| 3rd Beat | 60° | 60° | 100° | 100° |
| 4th Beat | 120° | 120° | 100° | 100° |

# Ⅲ. Program Penguin Bot to move forwards /backwards

The sketch used in this chapter is saved in below path and please refer to *Upload Penguin Bot program* and upload the codes.

*\ELEGOO Penguin Bot V2.0\Penguin Bot Function Introduction\Lesson 2 Make The Penguin Bot Move\Forward_Back\Forward_Back.ino*

## Code reviews:




```
#include "Oscillator.h"
#include <Servo.h>

#define YL_PIN 10
#define YR_PIN 9
#define RL_PIN 12
#define RR_PIN 6
#define N_SERVOS 4
Oscillator servo[N_SERVOS];

bool moveNServos(int time, int newPosition[])
{
  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)
  {
   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];
   }
   return false;
}

bool walk(int steps, int T, int dir)
{
   int move1[] = {90, 90 + 35, 90 + 15, 90 + 15};
   int move2[] = {90 + 25, 90 + 30, 90 + 15, 90 + 15};
   int move3[] = {90 + 20, 90 + 20, 90 - 15, 90 - 15};
   int move4[] = {90 - 35, 90, 90 - 15, 90 - 15};
   int move5[] = {90 - 40, 90 - 30, 90 - 15, 90 - 15};
   int move6[] = {90 - 20, 90 - 20, 90 + 15, 90 + 15};

   int move21[] = {90, 90 + 35, 90 - 15, 90 - 15};
   int move22[] = {90 + 25, 90 + 30, 90 - 15, 90 - 15};
   int move23[] = {90 + 20, 90 + 20, 90 + 15, 90 + 15};
   int move24[] = {90 - 35, 90, 90 + 15, 90 + 15};
   int move25[] = {90 - 40, 90 - 30, 90 + 15, 90 + 15};
   int move26[] = {90 - 20, 90 - 20, 90 - 15, 90 - 15};

   if (dir == 1) //Walking forward
   {
      for (int i = 0; i < steps; i++)
         if (
            moveNServos(T * 0.2, move1) || delays(50) ||
            moveNServos(T * 0.2, move2) || delays(50) ||
            moveNServos(T * 0.2, move3) || delays(100)  ||
            moveNServos(T * 0.2, move4) || delays(250) ||
            moveNServos(T * 0.2, move5) ||  delays(100) ||
            moveNServos(T * 0.2, move6) || delays(100))

            return true;
   }
   else //Walking backward
   {
      for (int i = 0; i < steps; i++)
        if ( moveNServos(T * 0.2, move21) || delays(50) ||
            moveNServos(T * 0.2, move22) || delays(50) ||
            moveNServos(T * 0.2, move23) || delays(100) ||
            moveNServos(T * 0.2, move24) || delays(250) ||
```

```
            moveNServos(T * 0.2, move25) || delays(100) ||
            moveNServos(T * 0.2, move26))
            return true;
    }
    return false;
}

void setup() {
  servo[0].attach(RR_PIN);
  servo[1].attach(RL_PIN);
  servo[2].attach(YR_PIN);
  servo[3].attach(YL_PIN);
}

void loop()
{
    for (int i = 0; i < 5; i++)
    {
        walk(1, 1500, 1);
    }
    for (int i = 0; i < 5; i++)
    {
        walk(1, 1500, -1);
    }
}
```

Execution: Call the walk function, change the value of the actual

In the above program, underlying program for servo drivers has been written and you would only need to use walk function to control Penguin Bot to walk forwards or backwards.

You need to set 3 parameters in Walk Function which are steps, T and dir.
steps: Set the walking steps of penguin Bot which in this case is 1
T: Set the walking speed of penguin Bot and unit is millisecond. -1 means walking backwards
dir: Set the walking direction of penguin Bot .1 means forwards and -1 means backwards

Therefore, codes above can make Penguin Bot walk forwards by 3 steps and then stop for 1 second and then walk backwards by 3 steps then stop. And keep recycling this way.

## IV. Program Penguin Bot to turn left/right

The sketch used in this chapter is saved in below path and please refer to *Upload Penguin Bot program* and upload the codes.

*\ELEGOO Penguin Bot V2.0\Penguin Bot Function Introduction\Lesson 2 Make The Penguin Bot*

*Move\ Right_Left\ Right_Left.ino*

## Code reviews:

```
#include "Oscillator.h"
#include <Servo.h>

#define YL_PIN 10
#define YR_PIN 9
#define RL_PIN 12
#define RR_PIN 6
#define N_SERVOS 4
Oscillator servo[N_SERVOS];
bool moveNServos(int time, int newPosition[])
{
   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)
   {
      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];
```

```
    }
    return false;
}
/*
    Turn control realization:
*/
bool turn(int steps, int T, int dir)
{
    int move1[] = {90 - 55, 90 - 20, 90 + 20, 90 + 20};
    int move2[] = {90 - 20, 90 - 20, 90 + 20, 90 - 20};
    int move3[] = {90 + 20, 90 + 55, 90 + 20, 90 - 20};
    int move4[] = {90 + 20, 90 + 20, 90 - 20, 90 + 20};
    int move5[] = {90 - 55, 90 - 20, 90 - 20, 90 + 20};
    int move6[] = {90 - 20, 90 - 20, 90 + 20, 90 - 20};
    int move7[] = {90 + 20, 90 + 55, 90 + 20, 90 - 20};
    int move8[] = {90 + 20, 90 + 20, 90 - 20, 90 + 20};

    int move21[] = {90 + 20, 90 + 55, 90 + 20, 90 + 20};
    int move22[] = {90 + 20, 90 + 20, 90 + 20, 90 - 20};
    int move23[] = {90 - 55, 90 - 20, 90 + 20, 90 - 20};
    int move24[] = {90 - 20, 90 - 20, 90 - 20, 90 - 20};
    int move25[] = {90 + 20, 90 + 55, 90 - 20, 90 + 20};
    int move26[] = {90 + 20, 90 + 20, 90 + 20, 90 - 20};
    int move27[] = {90 - 55, 90 - 20, 90 + 20, 90 - 20};
    int move28[] = {90 - 20, 90 - 20, 90 - 20, 90 - 20};

    if (dir == 1)
    {
        for (int i = 0; i < steps; i++)
            if (
                moveNServos(T * 0.2, move1) || delays(100) ||
                moveNServos(T * 0.2, move2) || delays(100) ||
                moveNServos(T * 0.2, move3) || delays(100) ||
                moveNServos(T * 0.2, move4) || delays(100) ||
                moveNServos(T * 0.2, move5) || delays(100) ||
                moveNServos(T * 0.2, move6) || delays(t / 5) ||
                moveNServos(T * 0.2, move7) || delays(100)   ||
                moveNServos(T * 0.2, move8) || delays(100))
                return true;
    }
    else
    {
        for (int i = 0; i < steps; i++)
            if (
```

```
            moveNServos(T * 0.2, move21) ||   delays(100) ||

            moveNServos(T * 0.2, move22) ||  delays(100) ||
            moveNServos(T * 0.2, move23) ||  delays(100) ||
            moveNServos(T * 0.2, move24) || delays(100) ||
            moveNServos(T * 0.2, move25) ||  delays(100) ||
            moveNServos(T * 0.2, move26) ||  delays(100) ||
            moveNServos(T * 0.2, move27) || delays(100) ||
            moveNServos(T * 0.2, move28) || delays(100))
            return true;
    }

    return false;
}

void setup()
{
    servo[0].attach(RR_PIN);
    servo[1].attach(RL_PIN);
    servo[2].attach(YR_PIN);
    servo[3].attach(YL_PIN);
    homes(100);
}

void loop()
{
    for (int i = 0; i < 5; i++)
    {
        turn(1, 2000, 1);
    }
    for (int i = 0; i < 5; i++)
    {
       turn(1, 2000, -1);
    }
}
```

In the above program, underlying program for servo drivers has been written and you would only need to use walk function to control Penguin Bot to walk forwards or backwards.

You need to set 3 parameters in Walk Function which are steps, T and dir.
steps: Set the walking steps of penguin Bot which in this case is 5
T: Set the walking speed of penguin Bot and unit is millisecond, which in this case is 2000ms
dir:Set the walking direction of penguin Bot .1 means turn right and -1 means turn left.

Therefore, codes above can make Penguin Bot turn right by 5 steps and then stop for 1 second and then turn left by 5 steps then stop. And keep recycling this way.