

```
totalhalt : first stop!  
  
one_way :  
turn : look to one side  
servo_turn : wait for the servo to be finished turning  
g 0, 0!  
totalhalt
```

```
the other way:  
turn : look to another side  
servo_turn : wait for the servo to be finished turning  
g 0, 0!  
totalhalt
```

```
so which is the better way?  
g2 : turn  
servo_turn  
  
back : turn
```

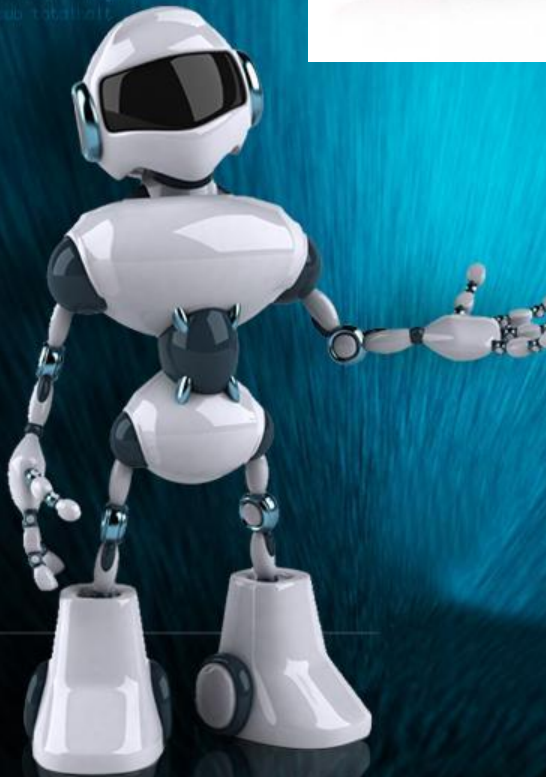
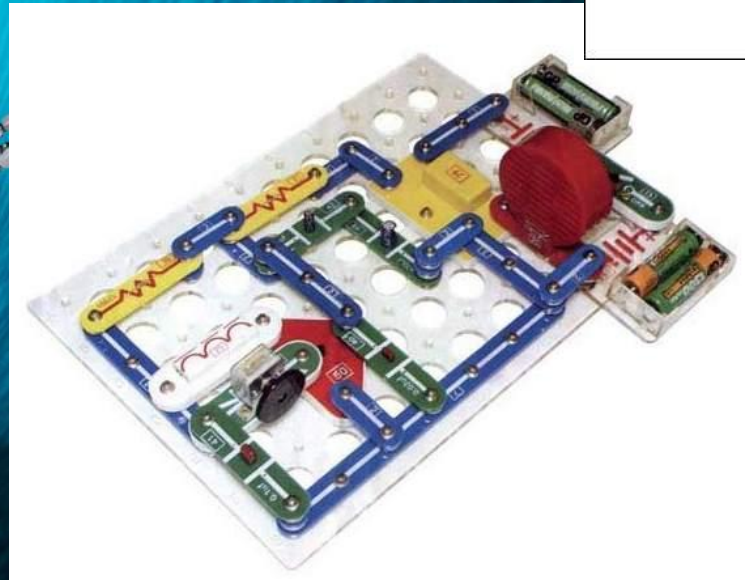
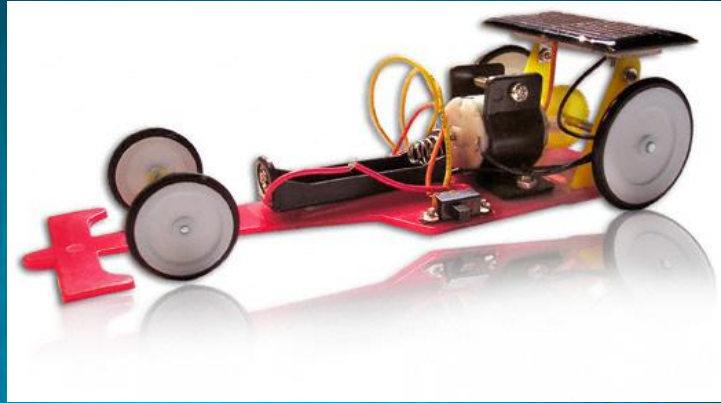
```
turns  
g : low 5 : low 7 : high 4  
turn : servo : totalhalt
```

```
light  
g : low 0 : low 2 : high 7  
turn : servo : totalhalt
```



# Перворобот

# Электронный конструктор



totalhalt : first stop  
back :  
turn : look to one side  
servo : turn & wait for the servo to be finished turning  
& go on  
totalhalt :

the other way :  
turn : look to another side  
servo : turn & wait for the servo to be finished turning  
& go on  
totalhalt :

on which is the better way :  
on : turn  
back : turn  
back : turn

turn :  
- low 5 : low 7 : high 4  
turn : servo : totalhalt :

turn :  
- low 6 : low 7 : high 7  
turn : servo : totalhalt :



```
totalhalt = first stop!  
brn = 0  
turn = look to one side  
servo.turn + wait for the servo to be finished turning  
g = 0  
totalhalt
```

```
the other way!  
turn = look to another side  
servo.turn + wait for the servo to be finished turning  
g = 0  
totalhalt
```

```
on which is the better way?  
on them  
back to turn
```

```
turns  
- low 5 : low 7 : high 4  
turn + servo totalhalt
```

```
turn  
- low 6 : low 7 : high 7  
turn + servo totalhalt
```

# Пластмассовый конструктор



# Металлический конструктор



[www.karusel-toys.ru](http://www.karusel-toys.ru)

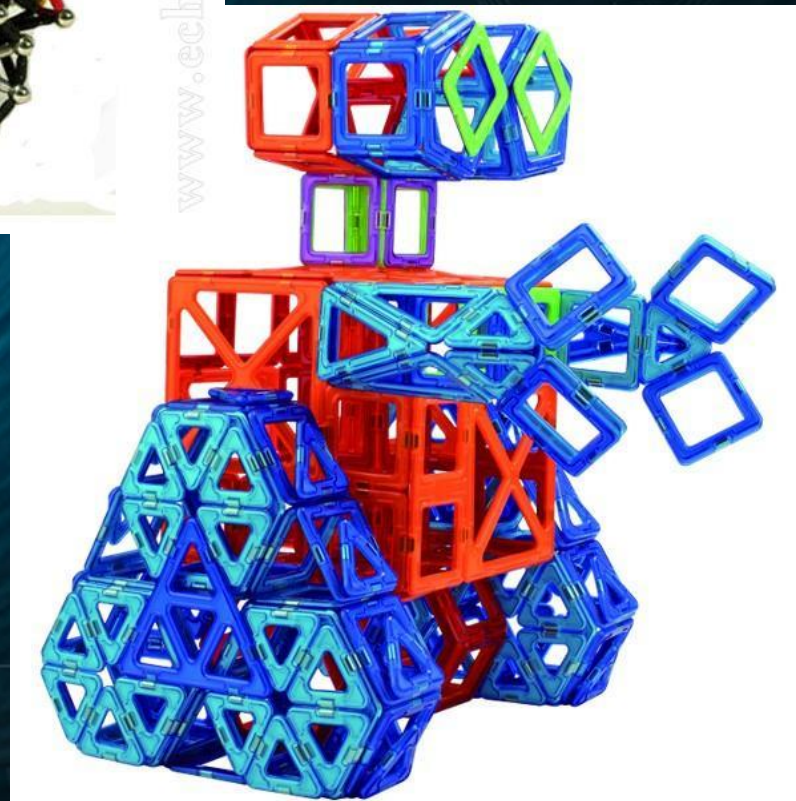




# Магнитный конструктор



www.echobot.ru



totalHeight = first.steel

break(){}

turn() look to one side

servo.turn() wait for the servo to be finished turning

g = 0; do

totalHeight

the other way:

turn() look to another side

servo.turn() wait for the servo to be finished turning

g = 0; do

totalHeight

so which is the better way:

do { turn()

servo.turn()

break() return

turns

low 5 : low 7 : high 4

turn() servo totalHeight

turns

low 6 : low 7 : high 7

turn() servo totalHeight

# Деревянный конструктор

totalHeight = first.steel

base.moveTo

turn() look to one side

servo.turn() wait for the servo to be finished turning

g.by() at

totalHeight

the other way:

turn() look to another side

servo.turn() wait for the servo to be finished turning

g.by() at

totalHeight

as which is the better way:

at then

base.turn

base.turn

turns

low 5 : low 7 : high 4

turn() servo totalHeight

turns

low 6 : low 7 : high 7

turn() servo totalHeight





# Конструктор из бумаги «Умная бумага»







# Конструктор Lego



totalhalt : first stop!  
back : wait  
turn : look to one side  
servo.turn : wait for the servo to be finished turning  
a : 90  
totalhalt!  
the other way!  
turn : look to another side  
servo.turn : wait for the servo to be finished turning  
- 90  
totalhalt!  
as which is the better way?  
a2 : turn  
back : turn  
back : turn  
turns  
- low 5 : low 7 : high 4  
turn : servo : totalhalt!  
high  
- low 6 : low 4 : high 7  
turn : servo : totalhalt!

# Спасение самолета

```
totalhalt : first stop!  
  
back : wait  
turn : look to one side  
servo.turn : wait for the servo to be finished turning  
g : 90 deg  
totalhalt
```

```
the other way:  
turn : look to another side  
servo.turn : wait for the servo to be finished turning  
g : 90 deg  
totalhalt
```

```
so which is the better way?  
a2 : yes!  
back : turn
```

```
back : turn
```

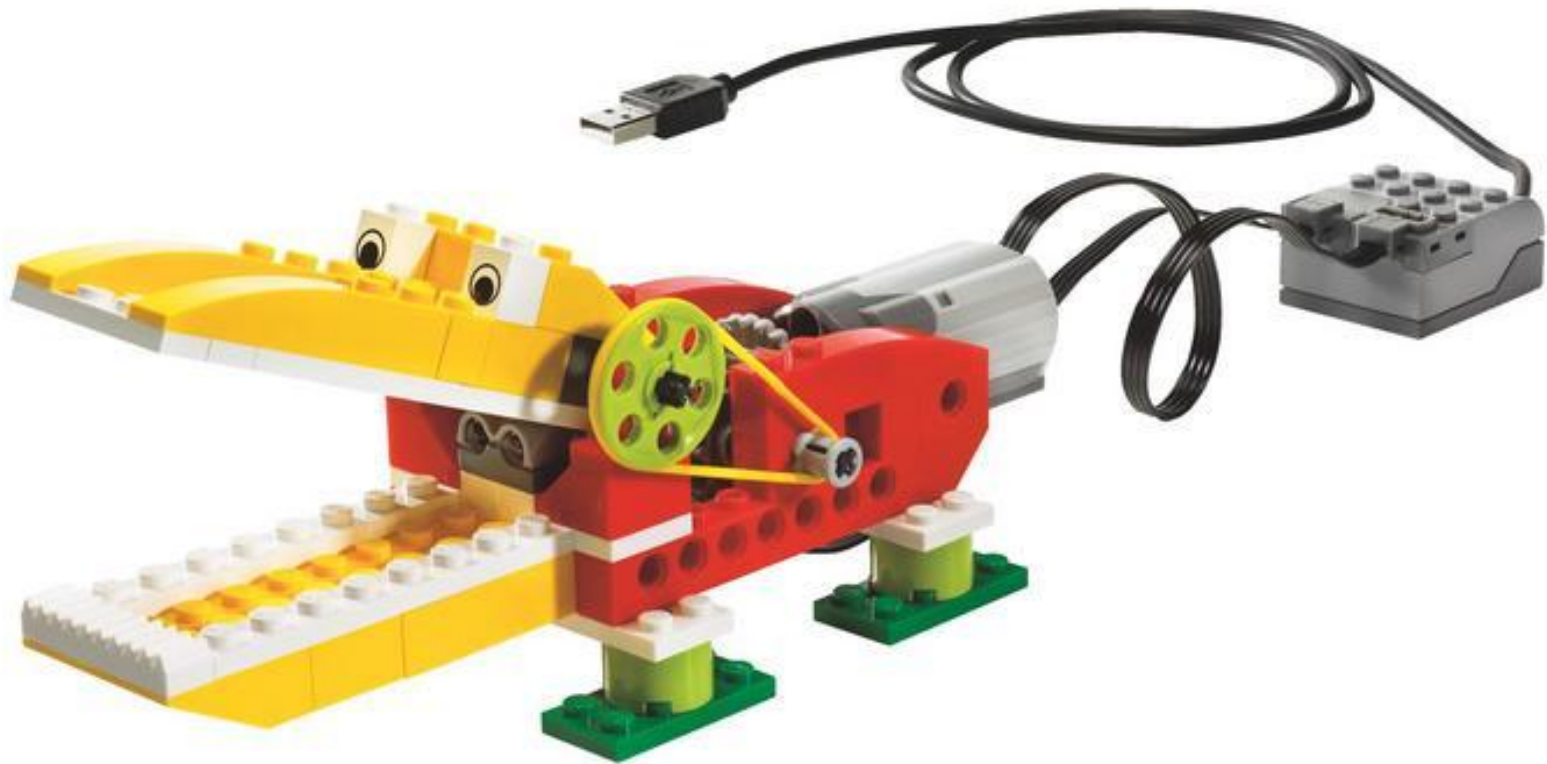
```
turns  
g : low 5 : low 7 : high 4  
turn : servo totalhalt
```

```
light  
g : low 6 : low 4 : high 7  
turn : servo totalhalt
```





# Голодный аллигатор



# Рычащий лев





# Порхающая птица



# Вратарь





# Танцующие ПТИЦЫ

```
totalHit = 0; first = 0;
bird = 0;
turn = 1; look to one side
servo.turn 1; wait for the servo to be finished turning
a = 0;
totalHit = 0;

the other way:
turn = 1; look to another side
servo.turn 1; wait for the servo to be finished turning
a = 0;
totalHit = 0;

be which is the better way?
a2 = 0;
bird = 0;
bird = 0;

turn = 1;
a = low 5 : low 7 : high 4
turn 1; servo totalHit;

bird = 1;
a = low 6 : low 7 : high 7
turn 1; servo totalHit;
```



# Обезьянка-барабанщица





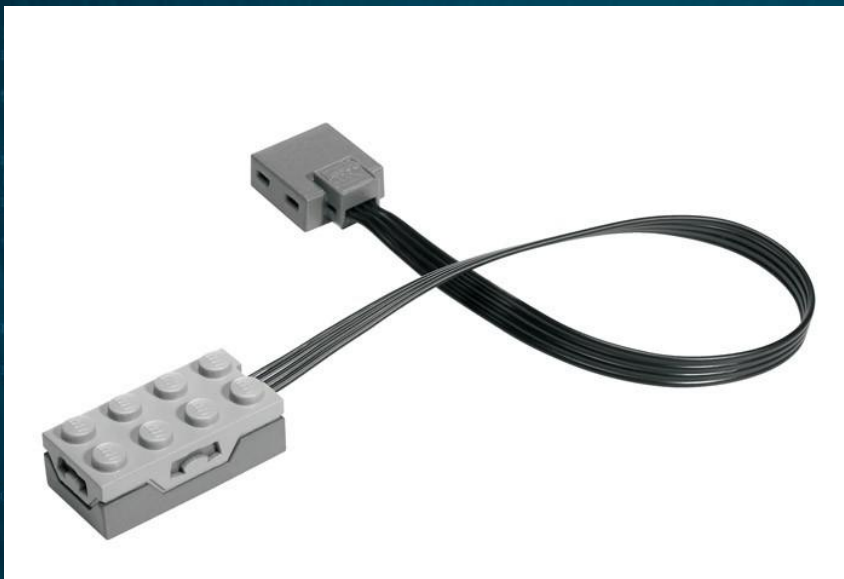
# Нападающий



# Конструктор Lego Wedo







**Датчик наклона**



**USB коммутатор**



**Датчик расстояния**



**Мотор**



# Конструктор NXT Mindstorm





totalhalt = 1 : first stop!

```
begin void  
  turn : look to one side  
  servo.turn : wait for the servo to be finished turning  
  g : 0, 0  
totalhalt
```

```
the other way!  
turn : look to another side  
servoturn : wait for the servo to be finished turning  
  g : 0, 0  
totalhalt
```

```
be which is the better way?  
a2 : turn  
b2 : turn  
b2 : turn
```

```
turns  
  : low 5 : low 7 : high 4  
turn : servo totalhalt
```

```
light  
  : low 0 : low 2 : high 7  
turn : servo totalhalt
```



```
totalhdlt : first stool
```

```
brk : hvt  
turn : look to one side  
servo_turn : wait for the servo to be finished turning  
g : 90  
totalhdlt
```

```
the other way:  
turn : look to another side  
servo_turn : wait for the servo to be finished turning  
g : 90  
totalhdlt
```

```
de : which is the better way:
```

```
ac : hvt  
brk : turn
```

```
brk : turn
```

```
turn :  
g : low 5 : low 7 : high 4  
turn : servo totalhdlt
```

```
turn :  
g : low 6 : low 7 : high 7  
turn : servo totalhdlt
```





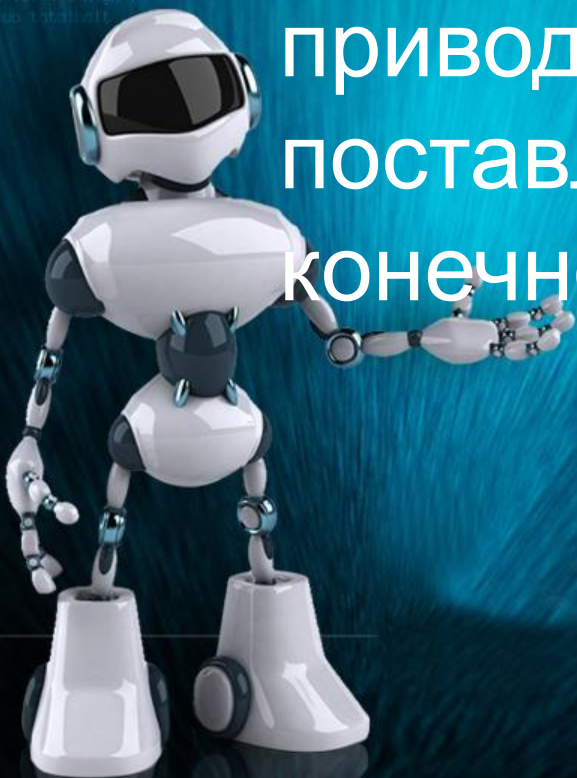
```
totalhalt = first_stop
loop:
  turn: look to one side
  servo_turn: wait for the servo to be finished turning
  g 0, 0
totalhalt
```

```
the other way:
turn: look to another side
servo_turn: wait for the servo to be finished turning
  g 0, 0
totalhalt
```

```
as which is the better way:
do: turn
servo_turn
loop: turn
totalhalt
```

```
turns
  low 5 : low 7 : high 4
turn: servo totalhalt
```

```
turns
  low 6 : low 4 : high 7
turn: servo totalhalt
```



# Алгоритм

- это описание последовательности действий, строгое исполнение которых приводит к решению поставленной задачи за конечное число шагов.