

```
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?  
a2 : yes!  
b2 : turn  
  
b2 : return
```

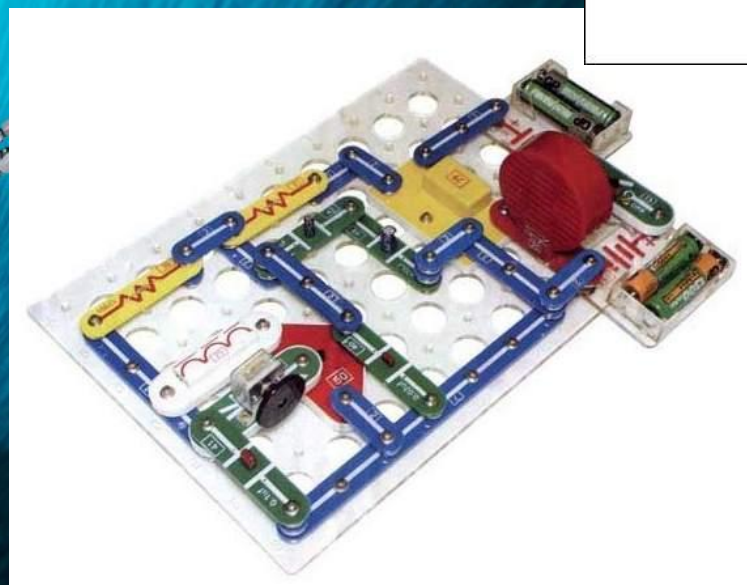
```
turns  
g low 5 : low 7 : high 4  
turn : servo : totalhalt
```

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



Перворобот

Электронный конструктор



bot.turn(1) : first stop
bot.wait
bot.turn(1) : look to one side
servo.turn(1) wait for the servo to be finished turning
bot.wait
bot.turn(1)

the other way:
bot.turn(1) : look to another side
servo.turn(1) wait for the servo to be finished turning
bot.wait
bot.turn(1)

or which is the better way:
bot.wait
bot.turn(1)

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

light
1 : low 6 : 1 : low 2 : high 7
turn : servo totalhalt

```
totalhalt = first.stoop
brn = low
turn = look to one side
servo.turn + wait for the servo to be finished turning
g = hi
totalhalt
```

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

```
on which is the better way:
a2 = turn
back.turn
```

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

```
turn
- low 6 : low 4 : high 7
turn + servo totalhalt
```

Пластмассовый конструктор



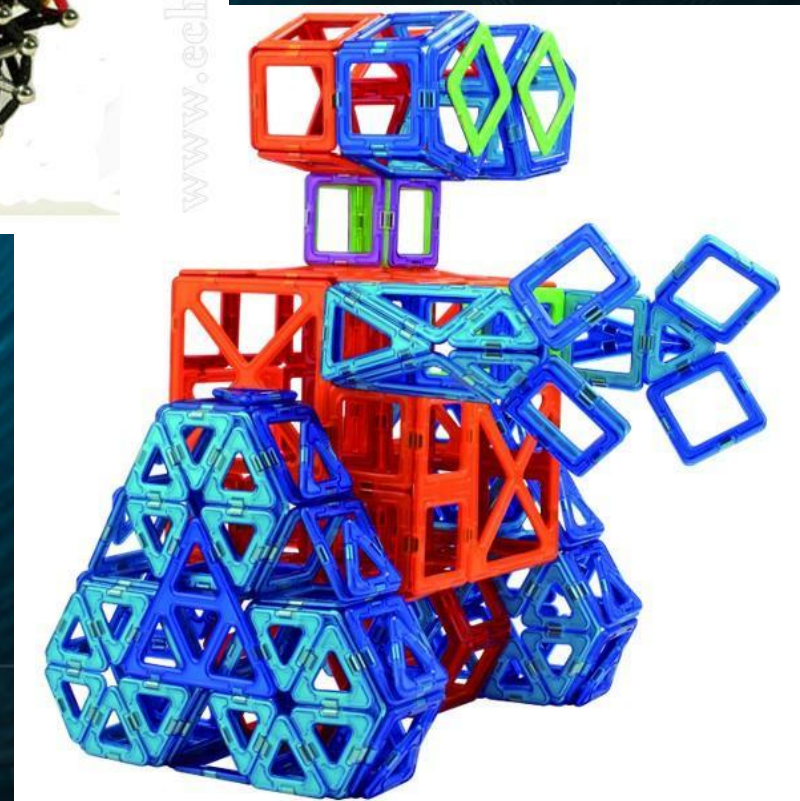
Металлический конструктор



www.karusel-toys.ru



Магнитный конструктор



```
totalhalt = first stop!  
  
low = 1  
high = 2  
servo.turn() wait for the servo to be finished turning  
a = 90  
totalhalt!  
  
the other way!  
return = look to another side  
servo.turn() wait for the servo to be finished turning  
a = 90  
totalhalt!  
  
as which is the better way?  
a2 = 180  
servo.turn()  
  
back = return  
  
turns  
a = low 5 : low 7 : high 4  
turn() servo totalhalt!  
  
high  
a = low 8 : low 9 : high 7  
turn() servo totalhalt!
```

Деревянный конструктор

```
totalHeight = first.steel  
brake.move()  
turn = look to one side  
servo.turn() wait for the servo to be finished turning  
g = g + 1  
totalHeight
```

```
the other way:  
turn = look to another side  
servo.turn() wait for the servo to be finished turning  
g = g + 1  
totalHeight
```

```
on which is the better way:  
do them  
brake.turn
```

```
brake.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»



Конструкторы из Бельгии Кликс



totalWait: 1000; stall: 1000;
begin: wait;
turn: look to one side
servo: turn; wait for the
60; do;
totalWait;

the other way:
turn: look to another side
servo: turn; wait for the servo to be finished turning
60; do;
totalWait;

as which is the better way:
do: turn;
servo: turn;

end: return;

turns:
- low 5; low 7; high 4
turn: servo; totalWait;

light:
- low 8; low 2; high 7
turn: servo; totalWait;

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 : 1 : 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
```

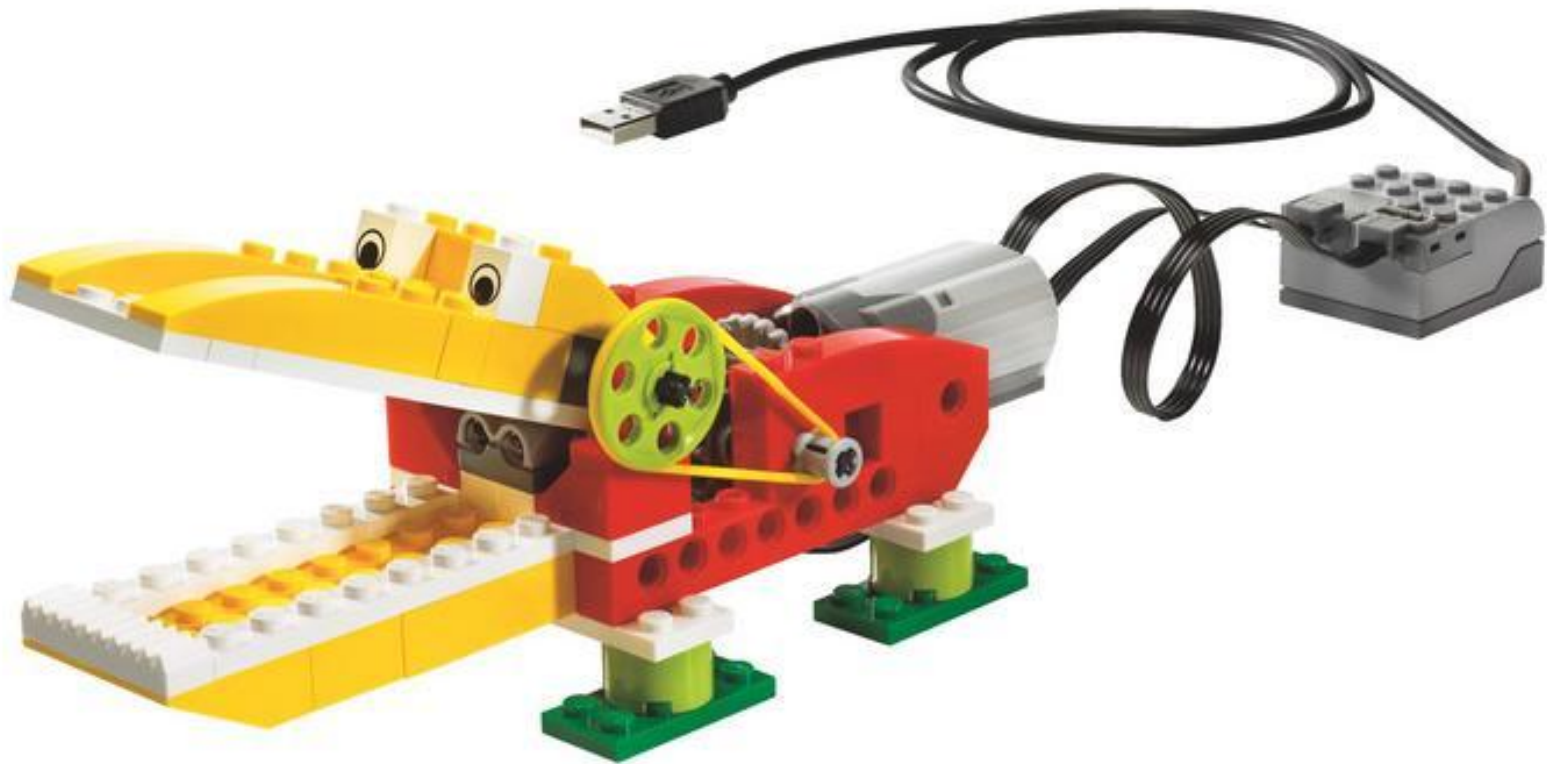
```
back.turn
```

```
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



```
def turn_right:
    turn(90)
    servo_turn = wait_for_the_servo_to_be_finished_turning
    return

def turn_left:
    turn(-90)
    servo_turn = wait_for_the_servo_to_be_finished_turning
    return
```

```
def turn_right:
    turn(90)
    servo_turn = wait_for_the_servo_to_be_finished_turning
    return
```

```
def turn_left:
    turn(-90)
    servo_turn = wait_for_the_servo_to_be_finished_turning
    return
```

```
def turn_right:
    turn(90)
    servo_turn = wait_for_the_servo_to_be_finished_turning
    return
```

Вратарь



Танцующие ПТИЦЫ

```
totalHalt = 0; first servo;
brn;
turn; // look to one side
servo.turn; // wait for the servo to be finished turning
p; // 10
totalHalt++;

the other way:
turn; // look to another side
servo.turn; // wait for the servo to be finished turning
p; // 10
totalHalt++;

be which is the better way?
p; // 10
totalHalt++;

turn;
p; // low 5 : low 7 : high 4
turn; // servo totalHalt;

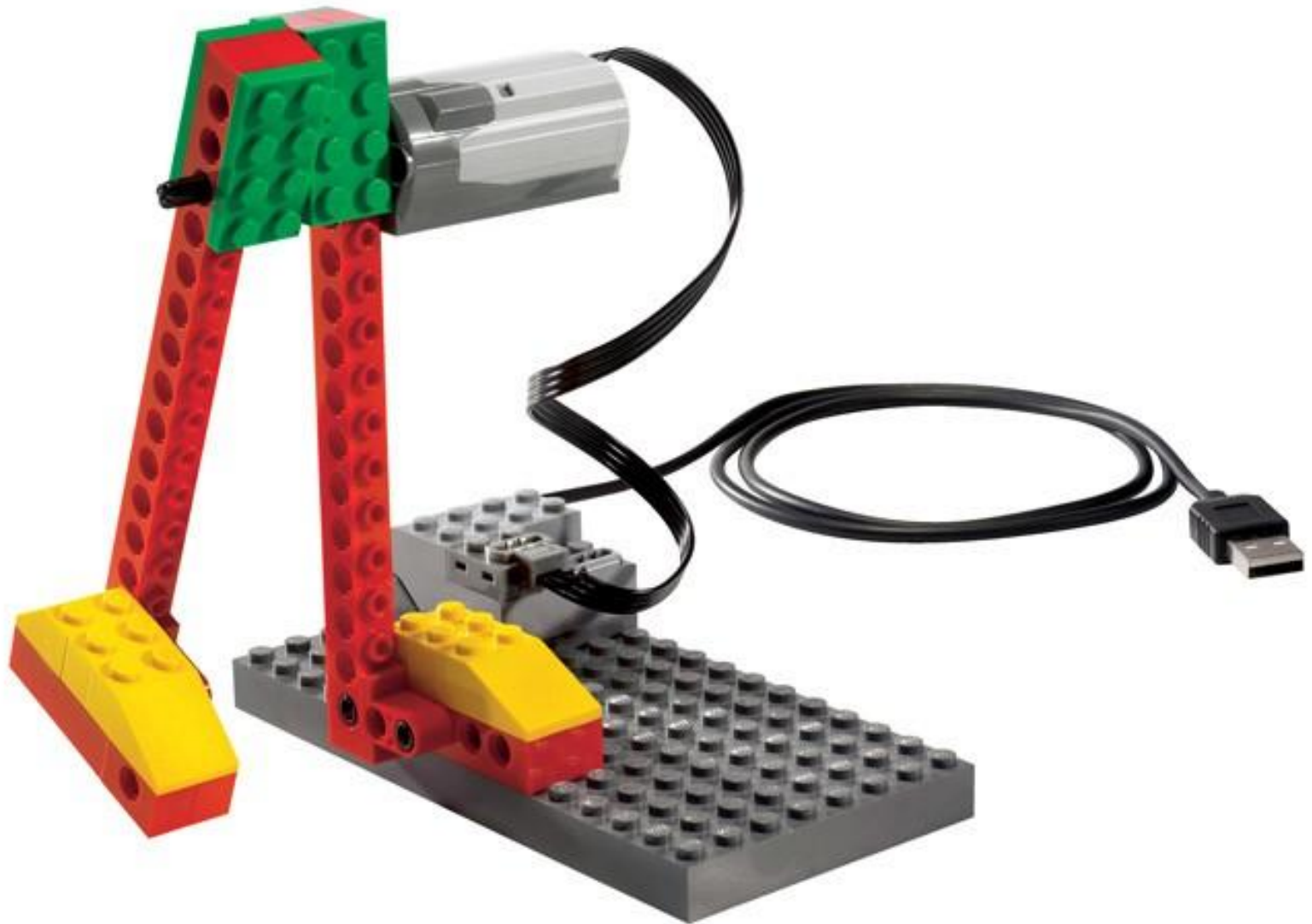
turn;
p; // low 6 : low 7 : high 7
turn; // servo totalHalt;
```



Обезьянка-барабанщица

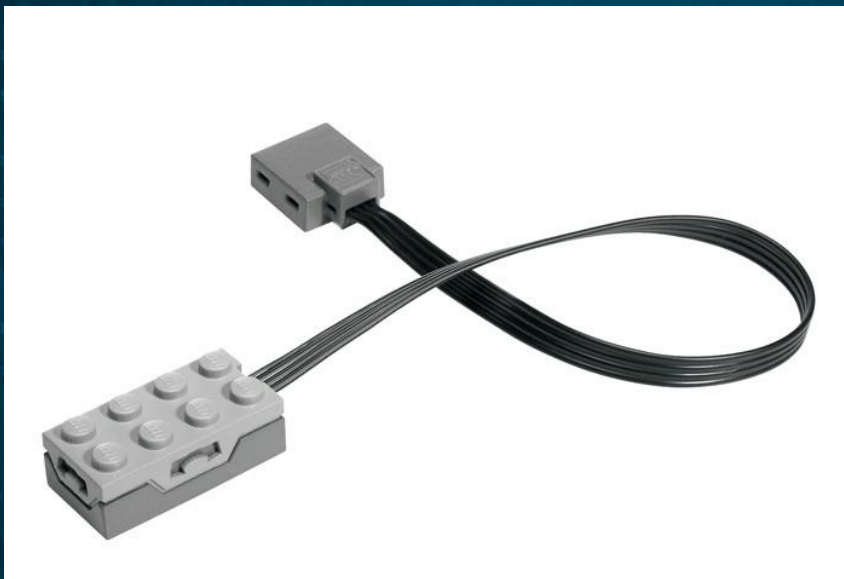


Нападающий



Конструктор Lego Wedo

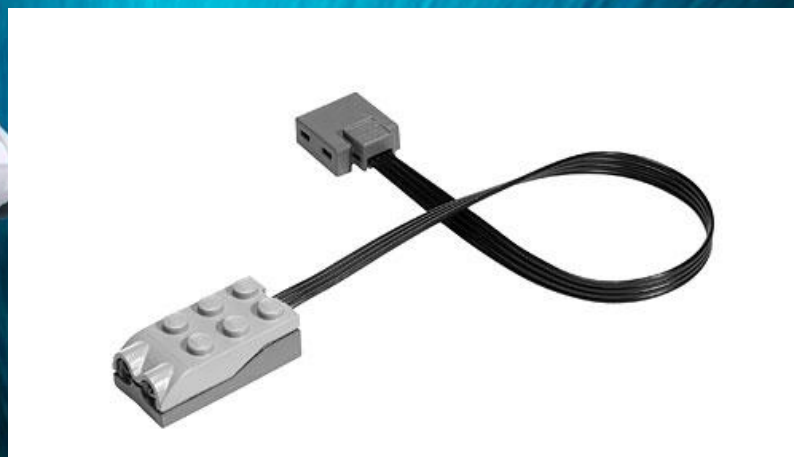




Датчик наклона



USB коммутатор



Датчик расстояния



Мотор



Конструктор NXT Mindstorm



totalhalt : first stop!

```
begin 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  
servoturn : wait for the servo to be finished turning  
  g : 0, 0  
totalhalt
```

```
be which is the better way:  
a2 : turn  
back : turn  
back : turn
```

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

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



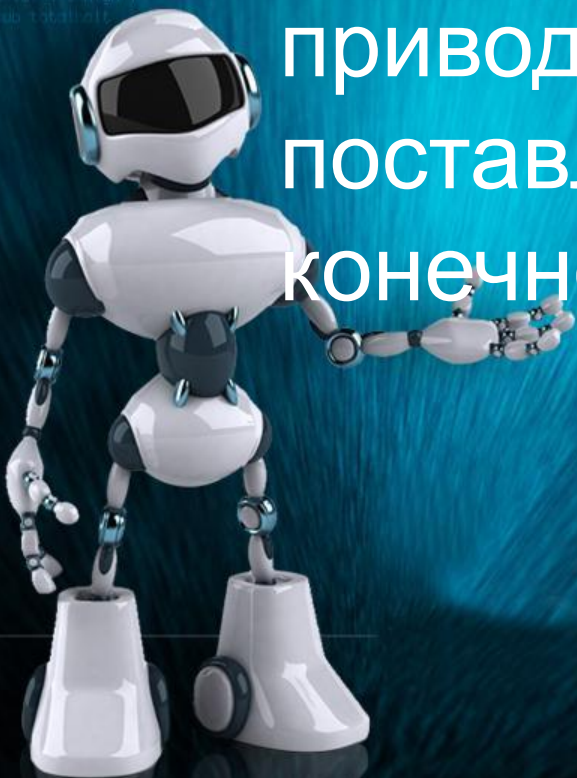

```
totalhalt = first_stop
turn = 1
servo.turn(1) wait for the servo to be finished turning
g = 0; do
totalhalt
```

```
the other way:
turn = 1 look to another side
servo.turn(-1) wait for the servo to be finished turning
g = 0; do
totalhalt
```

```
do which is the better way:
do them
servo.turn
end return
```

```
turns
low 5 : low 7 : high 4
turn 1 servo totalhalt
```

```
turns
low 6 : low 4 : high 7
turn 1 servo totalhalt
```



Алгоритм

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