

**ГОУ ДОД
Областной Центр детского (юношеского) технического творчества**

**Всероссийский этап Международных Состязаний Роботов 2012
Средняя возрастная категория**

Описание проекта

«Вестник будущего»

Выполнили:
Захарчев Александр
Зайцев Ростислав
Руководитель проекта:
Горшков Борис Константинович

**г. Ульяновск
2012 г.**

Пояснительная записка

Проект был создан для реализации темы творческой категории состязаний роботов по международным правилам 2012 г. в средней возрастной категории.

Тема WRO 2012 для творческой категории – «Роботы объединяют людей».

Проект реализован на базе процессоров NXT.

Интеллектуальный роботизированный комплекс решения игровых задач «Вестник будущего»

Проблема:

Одним из специфических лишь для человека видов деятельности являются настольные интеллектуальные игры. Робот, способный соперничать с человеком в этом занятии, отражал бы одну из важных сторон природы человека.

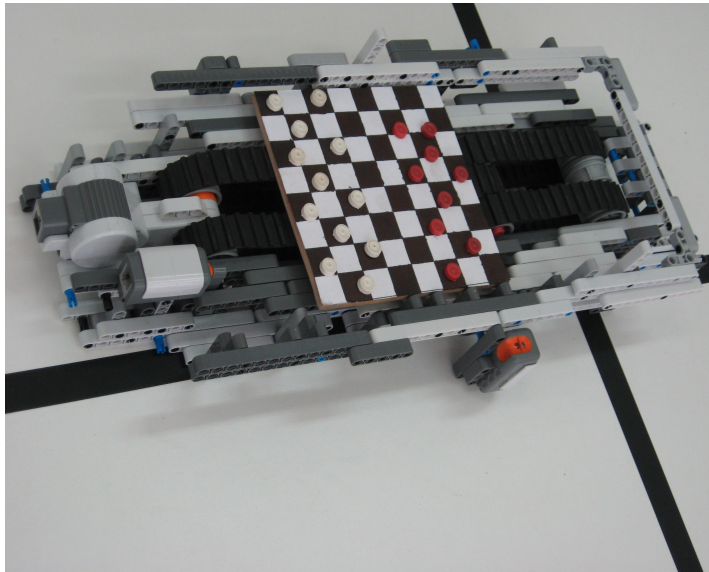
Цель:

Создать робота, способного играть с человеком в настольные интеллектуальные игры. Для этого:

1. Создать механическую основу для анализа игровой ситуации на поле и манипулирования игровыми элементами.
2. Разработать алгоритмы для определенных операций воздействия на поле.
3. Создать алгоритм принятия решений для какой-либо настольной игры(в качестве такой игры были выбраны русские шашки).

Робот состоит из следующих блоков:

1. Платформа с подвижной лентой:



Предназначена для размещения на ней игрового поля.
Обеспечивает перемещение поля относительно сканера и захвата по оси OY.

2. Захват и сканер.



Перемещаясь по перпендикулярной балке сканер и захват движутся относительно поля вдоль оси ОХ. Сканер производит обход игрового поля и анализ текущей ситуации. Захват служит для перемещения шашек по полю.

Робот реализован с использованием:

1	1 Микрокомпьютер NXT
2	3 сервомотора
3	6 проводов
4	Датчик касания
5	Цветовой датчик
6	Световой датчик
7	Набор деталей NXT 2.0

Пример программы

```
//#define TRUE false
#define dx 47
#define dy 1500
#define dz 35

#define diagonalspeed 15

int x;
int y;
int board[100];

void fingerDown(){ //Opuskaem palec vniz, zahvativaya shashku
    RotateMotor(OUT_B,25,dz) ;
}

void fingerUp(){ //Opuskaem palec vverx, otpuskaya shashku
    RotateMotor(OUT_B,-25,dz) ;
}

void fingerRight(){ //Dvigaem palec vpravo, dvigaya shashku
    RotateMotor(OUT_C,-50, dx) ;
    // Wait(100);
    x++ ;
}

void fingerLeft(){ //Dvigaem palec vlevo, dvigaya shashku
    RotateMotor(OUT_C,50, dx) ;
    // Wait(100);
    x-- ;
}

void fingerForvard(){ //Dvigaem palec vperet, dvigaya shashku
    RotateMotor(OUT_A,50,dy) ;
    // Wait(100);
    y++;
    Off(OUT_A);
}

}
```

```

void fingerBack(){ //Dvigaem palec nazad, dvigaya shashku
    RotateMotor(OUT_A,-50,dy) ;
    // Wait(100);
    y--;
    Off(OUT_A);
}
void Back1(){ //Dvigaem palec NAZAD, dvigaya shashku
    RotateMotor(OUT_C,50, dx * 7) ;
    // Wait(100);
    x-=7;
}
void Back2(){ //Dvigaem pole NAZAD, dvigaya shashku
    RotateMotor(OUT_A,50,dy*7) ;
    // Wait(100);
    y-=7;
}

```

```

void FL(){ //Dvigaem palec vpravo i vpered, dvigaya shashku
    x--;
    y++;
    OnFwdEx(OUT_C,diagonalspeed,RESET_ALL);
    OnFwdEx(OUT_A,127,RESET_ALL);
    Wait(100);
    while ((GetOutput(OUT_C,TachoCount) < dx) ||
(GetOutput(OUT_A,TachoCount) < dy) ){
        if((GetOutput(OUT_C,TachoCount) > dx)){

            OffEx(OUT_C,RESET_NONE);
        }
        if((GetOutput(OUT_A,TachoCount)>dy)){
            OffEx(OUT_A,RESET_NONE);
        }
    }
}

```

```

Off(OUT_A);
Off(OUT_C);
}

void BR(){ //Dvigaem palec nazad i vlevo, dvigaya shashku
    x++;
    y--;
    OnFwdEx(OUT_C,-diagonalspeed,RESET_ALL);
    OnFwdEx(OUT_A,-127,RESET_ALL);
    Wait(100);
    while ((GetOutput(OUT_C,TachoCount) > -dx) ||
(GetOutput(OUT_A,TachoCount) > -dy) ){
        if((GetOutput(OUT_C,TachoCount) < -dx)){
            OffEx(OUT_C,RESET_NONE);
        }
        if((GetOutput(OUT_A,TachoCount)<-dy)){
            OffEx(OUT_A,RESET_NONE);
        }
    }
    Off(OUT_A);
    Off(OUT_C);
}

```

```

void FR(){ //Dvigaem palec vpravo i vpered, dvigaya shashku
    x++;
    y++;
    OnFwdEx(OUT_C,-diagonalspeed,RESET_ALL);
    OnFwdEx(OUT_A,127,RESET_ALL);
    Wait(100);
    while ((GetOutput(OUT_C,TachoCount) > -dx) ||
(GetOutput(OUT_A,TachoCount) <dy) ){
        if((GetOutput(OUT_C,TachoCount) < -dx)){

            OffEx(OUT_C,RESET_NONE);
        }
        if((GetOutput(OUT_A,TachoCount)> dy)){
            OffEx(OUT_A,RESET_NONE);
        }
    }
}

```



```

}
// 1455 ,2413

Off(OUT_A);
Off(OUT_C);
}
void BL(){ //Dvigaem palec nazad i vpravo, dvigaya shashku
    x--;
    y--;
    OnFwdEx(OUT_C,diagonalspeed,RESET_ALL);
    OnFwdEx(OUT_A,-127,RESET_ALL);
    Wait(100);
    while ((GetOutput(OUT_C,TachoCount) < dx) ||
(GetOutput(OUT_A,TachoCount) > -dy) ){
        if((GetOutput(OUT_C,TachoCount) > dx)){

            OffEx(OUT_C,RESET_NONE);
        }
        if((GetOutput(OUT_A,TachoCount)< -dy)){
            OffEx(OUT_A,RESET_NONE);
        }
    }
}

Off(OUT_A);
Off(OUT_C);
}
void Zero() { //Vivod na nol dvigaya shashku
    x=1;
    y=1;
    OnFwd(OUT_C,-50) ;
    SetSensorColorBlue(IN_3);
    NumOut(50, 10, SensorRaw(IN_4), DRAW_OPT_NORMAL);
    while(SensorRaw(IN_4) >200){ NumOut(50, 10, SensorRaw(IN_4),
DRAW_OPT_NORMAL);};
    SetSensorColorFull(IN_3);
    Off(OUT_C) ;
    OnFwd(OUT_A,-100) ;
    while(!Sensor(IN_2)){ } ;

```

```

Off(OUT_A)          ;
RotateMotor(OUT_A,100,2100) ;
RotateMotor(OUT_C,50,500) ;
}
void scan_mode(){
    RotateMotor(OUT_A,-100,2 *dy) ;
    RotateMotor(OUT_C,-50, 35) ;

}
void work_mode(){
    RotateMotor(OUT_A,100,2 * dy) ;
    RotateMotor(OUT_C,50, 15) ;
}
void Scan(){ //Scaniruem pole, dvigaya shashku
    for(int i = 1; i <= 8; i++)
    {
        for(int j = 1; j <= 8; j++)
        {
            board[i*10+j] = Sensor(IN_3);
            NumOut(50, 50, board[i*10+j], DRAW_OPT_NORMAL);
            NumOut(40, 50, Sensor(IN_3), DRAW_OPT_NORMAL);
            NumOut(60, 50, i, DRAW_OPT_NORMAL);
            NumOut(70, 50, j, DRAW_OPT_NORMAL);
            switch (board[i*10+j]) {
                case 1:
                    PlayTone(440,0.3);
                    RectOut(i*4 + 1 , j* 4 + 1 , 2,2, DRAW_OPT_NORMAL);
                    break;

                case 5:
                    CircleOut(i*4 + 2, j* 4 + 2, 1, DRAW_OPT_NORMAL); break;

                default:PlayTone(220, 0.3);
                    RectOut(i*4 , j* 4 , 4,4, DRAW_OPT_NORMAL);

            }
        }
        if( j < 8) {

            fingerRight();

```

```

        Wait(300);
    } else { Back1() ;
fingerForvard() ;}
    }

}
}

void moveToXY(int nx, int ny){
    if(x<nx){
        for(int i=x; i<nx; i++) {
            fingerRight();
        }

    }
    if(x>nx){
        for(int i=x; i>nx; i--) {
            fingerLeft() ;
        }

    }
    if(y<ny){
        for(int i=y; i<ny; i++) {
            fingerForvard() ;
        }

    }
    if(y>ny){
        for(int i=y; i>ny; i--) {
            fingerBack() ;
        }

    }
}

void hit(int x1, int y1, int x2, int y2){
    moveToXY(x1 + (x2-x1)/2 ,y1 + (y2-y1)/2);
    fingerDown();
    fingerRight();
}

```

```

while( x < 8 ){
    FR() ;
    Wait(100) ;
}
fingerUp();
moveToXY(x1,y1);
fingerDown();
if (x2 > x1){
    if (y2 > y1){
        FR();
        FR();
    }
    else{
        BR();
        BR();
    }
}
else{
    if (y2 > y1){
        FL();
        FL();
    }
    else{
        BL();
        BL();
    }
}
fingerUp();
}

```

```

task coordOut() { //Koordinati
    while (true){
        NumOut(50, 30, x, DRAW_OPT_NORMAL);
        NumOut(65, 30, y, DRAW_OPT_NORMAL);
        Wait(500);
    }
}

```

```
}  
}
```

```
task main(){  
    start coordOut;  
    SetSensor(IN_4, SENSOR_LIGHT);  
    SetSensorTouch(IN_2);  
    SetSensorColorFull(IN_3);  
    Zero();  
    Scan();  
  
}
```