

ДОДАТОК А

Лістинг програми

```
// Файл App.h

#pragma once

#include <afxwin.h>
#include <afxcmn.h>
#include <afxmt.h>

class CApp : public CWinApp
{
public:
    BOOL InitInstance(void);
};

class CSDialog : public CDialog
{
    CButton *ch1,*rb1,*rb2,*rb3,*rb4;
    CEdit *ed1,*ed2,*ed3;
    CEdit *LPath;
    CSliderCtrl *s11;
public:
    CSDialog(UINT id,CWnd *Owner):CDialog(id,Owner){}
    BOOL OnInitDialog();
    void OnForward();
    void OnBack();
    void OnRight();
    void OnLeft();
    void OnStop();
    void OnMove();
    void OnGripperOpen();
    void RotLeft(double angle);
    void RotRight(double angle);
    void OnWay();
    DECLARE_MESSAGE_MAP()
};
```

```
// Файл App.cpp

#include "App.h"
#include "resource.h"

#include <cstdlib>
#include <iostream>
#include <string>
#include <conio.h>
#include <math.h>
#include "nxt.h"

using namespace std;
double pi = 3.1415926535;

//set up the NXT
Connection *connection = new Bluetooth();
Sensor *sensor1 = new Touch(IN_1, connection);
Sensor *sensor2 = new Sonar(IN_4,connection);
Motor *motorA = new Motor(OUT_A, connection);
Motor *motorB = new Motor(OUT_B, connection);
```

```

Motor *motorC = new Motor(OUT_C, connection);

BOOL CApp::InitInstance(void)
{
    CSDialog a(IDD_DIALOG1,NULL);
    a.DoModal();
    connection->disconnect();
    return TRUE;
}

CApp App;

BEGIN_MESSAGE_MAP(CSDialog, CDialog)
    ON_COMMAND(IDC_BUTTON1,OnForward)
    ON_COMMAND(IDC_BUTTON2,OnBack)
    ON_COMMAND(IDC_BUTTON3,OnLeft)
    ON_COMMAND(IDC_BUTTON4,OnRight)
    ON_COMMAND(IDC_BUTTON5,OnStop)
    ON_COMMAND(IDC_BUTTON6,OnMove)
    ON_COMMAND(IDC_BUTTON7,OnWay)
    ON_BN_CLICKED(IDC_CHECK1,OnGripperOpen)
END_MESSAGE_MAP()

BOOL CSDialog::OnInitDialog()
{CDialog::OnInitDialog();
    InitCommonControls();
    ch1=(CButton*)GetDlgItem(IDC_CHECK1);
    ed1=(CEdit*)GetDlgItem(IDC_EDIT1);
    ed2=(CEdit*)GetDlgItem(IDC_EDIT2);
    ed3=(CEdit*)GetDlgItem(IDC_EDIT3);
    LPath = (CEdit*)GetDlgItem(IDC_EDIT4);
    rb1=(CButton*)GetDlgItem(IDC_RADIO1);
    rb2=(CButton*)GetDlgItem(IDC_RADIO2);
    rb3=(CButton*)GetDlgItem(IDC_RADIO3);
    rb4=(CButton*)GetDlgItem(IDC_RADIO4);
    s11=(CSliderCtrl*)GetDlgItem(IDC_SLIDER1);
    s11->SetPos(25);
    ch1->SetCheck(0);rb1->SetCheck(1);
    char a[255];
    try{
        wsprintf(a,"Try to connect to the NXT");
        connection->connect(18);
        MessageBox("Connected");
    }
    catch (Nxt_exception& e){
        MessageBox(e.what());
        connection->disconnect();
        EndDialog(1);
    }

    return TRUE;
}

void CSDialog::OnForward()
{
    motorB->on(s11->GetPos());
    motorC->on(s11->GetPos());
    int SensorValue = sensor2->read();
    char str[3]="";
    wsprintf(str,"%d",SensorValue);
    LPath->SetWindowText(str);
}

void CSDialog::OnBack()

```

```

{motorB->on(-s11->GetPos());
motorC->on(-s11->GetPos());
//this->GetParentFrame()->Invalidate(FALSE);
}

void CSDialog::OnLeft()
{motorB->on(s11->GetPos());
motorC->on(0);
//this->GetParentFrame()->Invalidate(FALSE);
}

void CSDialog::OnRight()
{motorB->on(0);
motorC->on(s11->GetPos());
//this->GetParentFrame()->Invalidate(FALSE);
}

void CSDialog::OnStop()
{motorB->on(0);
motorC->on(0);
//this->GetParentFrame()->Invalidate(FALSE);
}

void CSDialog::OnGripperOpen()
{if(ch1->GetCheck())
{motorA->on(s11->GetPos());
//Sleep(50);
}
else motorA->on(-s11->GetPos());
//this->GetParentFrame()->Invalidate(FALSE);
}

void CSDialog::OnMove()
{
double Xi=0, TP=0, angle=0, Xv=0, Vi=0;
char a[255];
ed1->GetWindowText(a,5);
TP = atoi(a);
double curPos=0;
while(curPos < TP)
{
CString b;
double L=pi*55;// L- dlina okrugnosti, 55mm d - diametr kola v mm
if(curPos < 0.2*TP)
for(int i=1; i<11; i++)
{
Xi=(0.2*TP*i)/10; // tekushee pologenie robota, TP - target Point
double Xa=0.02*TP;
angle = 360*Xa/L; // ugol povorota kola
Xv=0.02*i;
Vi = (1.8*sqrt(Xv)*100)/1.5; //vichislenie skorosti vrashenia motorov
b.Format(_T("%f"),Vi);
SetWindowText(b);
motorB->on(Vi,angle);
motorC->on(Vi,angle);
Sleep((int)Xa/Vi*220);
curPos = Xi;
}
if(curPos>=0.2*TP && curPos<0.8*TP)
{
double Xa=0.6*TP;
angle = 360*Xa/L;
Vi = 60;
}
}
}

```

```

        b.Format(_T("%f",Vi));
        SetWindowText(b);
        motorB->on(Vi,angle);
        motorC->on(Vi,angle);
        Sleep((int)Xa/Vi*220);
        curPos = 0.801*TP;
    }

    if (curPos>=0.8*TP)
        for(int i=1; i<11; i++)
        {
            Xi=((0.2*TP*i)/10)+0.8*TP;
            double Xa=0.02*TP;
            angle= 360*Xa/L;
            Xv=0.02*i+0.8;
            Vi=(-20*Xv*Xv+32*Xv-12)*100/1.5;
            //b.Format(_T("%f",Vi));
            //SetWindowText(b);
            motorC->on(Vi,angle);
            motorB->on(Vi,angle);
            Sleep((int)Xa/Vi*220);
            curPos = Xi;
        }
    }
}

void CSDialog::RotLeft(double angle)
{
    motorB->on(0,angle);
    motorC->on(40,angle);
    Sleep(1000);
}

void CSDialog::RotRight(double angle)
{
    motorB->on(40,angle);
    motorC->on(0,angle);
    Sleep(1000);
}

void CSDialog::OnWay()
{
    double curX=0.0, curY=0.0, curAng=90.0, tarAng=0.0, newAng=0.0, rotAng=0.0;

    char s2[25],s3[25];
    ed2->GetWindowText(s2,strlen(s2));
    ed3->GetWindowText(s3,strlen(s3));
    double tarX=atof(s2);
    double tarY=atof(s3);

    double a = tarX - curX;
    double b = tarY - curY;
    double c = sqrt(a*a + b*b);
}

```

```

if (tarX>curX && tarY>=curY)
{
    double sinAng = a/c;
    tarAng=asin(sinAng)*57.2168576;
    newAng= 90.0 - curAng - tarAng;
    rotAng = abs(4.07*newAng);
    if (newAng>=0)RotLeft(rotAng);
    else RotRight(rotAng);
    Sleep(100);
}

if (tarX>=curX && tarY<curY)
{
    double sinAng = a/c;
    tarAng=asin(sinAng)*57.2168576;
    newAng= 270.0 - curAng + tarAng;
    rotAng = 4.07*newAng;
    if (newAng>=0)RotLeft(rotAng);
    else RotRight(rotAng);
    Sleep(100);
}

if (tarX<curX && tarY<=curY)
{
    double sinAng = a/c;
    tarAng=asin(sinAng)*57.2168576;
    newAng= 270.0 - curAng + tarAng;
    rotAng = 4.07*newAng;
    if (newAng>=0)RotLeft(rotAng);
    else RotRight(rotAng);
    Sleep(100);
}

if (tarX<=curX && tarY>curY)
{
    double sinAng = a/c;
    tarAng=asin(sinAng)*57.2168576;
    newAng= 90.0 - curAng - tarAng;
    rotAng = 4.07*newAng;
    if (newAng>=0)RotLeft(rotAng);
    else RotRight(rotAng);
    Sleep(100);
}

curX=tarX;
curY=tarY;
curAng=newAng;
}

```

ДОДАТОК Б
Демонстраційний матеріал

